Efficient Low-Resource Language Adaptation via Multi-Source Dynamic Logit Fusion

Chen Zhang, Jiuheng Lin, Zhiyuan Liao, Yansong Feng*
Wangxuan Institute of Computer Technology, Peking University
{zhangch,fengyansong}@pku.edu.cn
{linjiuheng,liaozy}@stu.pku.edu.cn

Abstract

Adapting large language models (LLMs) to low-resource languages (LRLs) is constrained by the scarcity of task data and computational resources. Although Proxy Tuning offers a logit-level strategy for introducing scaling effects, it often fails in LRL settings because the large model's weak LRL competence might overwhelm the knowledge of specialized smaller models. We thus propose TRIMIX, a test-time logit fusion framework that dynamically balances capabilities from three different sources: LRL competence from a continually pretrained small model, task competence from high-resource language instruction tuning, and the scaling benefits of large models. It is data- and compute-efficient, requiring no LRL task annotations, and only continual pretraining on a small model. Experiments across four model families and eight LRLs show that TRIMIX consistently outperforms single-model baselines and Proxy Tuning. Our analysis reveals that prioritizing the small LRL-specialized model's logits is crucial for success, challenging the prevalent large-model-dominant assumption.

1 Introduction

Although large language models (LLMs) have achieved remarkable success in high-resource languages (HRLs), their ability to process low-resource languages (LRLs) remains limited (Singh et al., 2025). To improve the LLM performance in LRLs, researchers attempt to adapt the HRL-dominant LLMs to LRLs (Ke et al., 2025), which typically face two fundamental challenges. One challenge is the scarcity of labeled task data for these languages (Joshi et al., 2020), due to the high cost of manual annotation. Consequently, there is insufficient fine-tuning data to endow models with task-solving capabilities. The other challenge lies in the scarcity of computational budget (Urbizu et al., 2025; Hernández et al., 2025). Researchers working on LRLs often rely on smaller models and lack the resources required to continually pretrain large-scale models, even though larger models typically exhibit stronger general abilities.

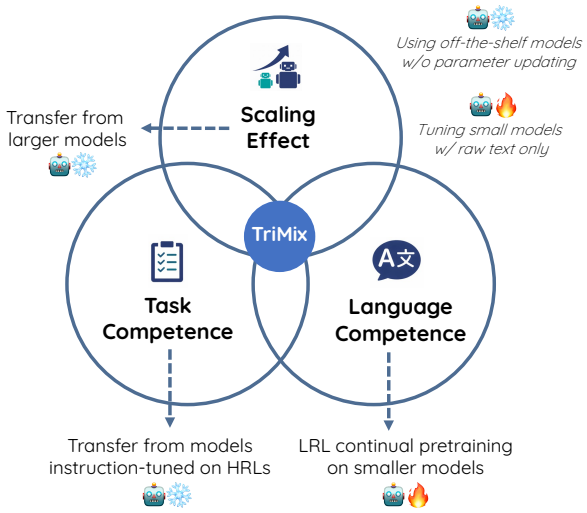

Figure 1: TRIMIX integrates three sources of benefit for LRL adaptation while minimizing the need for annotating task data and tuning larger models.

Recent studies have explored paradigms avoiding reliance on annotated LRL task data, such as model merging (Tao et al., 2024; Yamaguchi et al., 2025). It combines task-solving capabilities learned from HRLs with language competence acquired through continual pretraining (CPT) on the target LRL. However, model merging requires the merging models to share the same architecture and scale; consequently, scaling up still necessitates CPT on a larger model, which is computationally expensive. To alleviate computational constraints, Proxy Tuning (Liu et al., 2024) aims to approximate the benefits of training a larger specialized model by injecting the knowledge of a smaller, domain-adapted model into the logits of a larger one. This approach shows promising results in the code domain. However, applying it to LRL set-

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4540–4557

tings is non-trivial. In such scenarios, the larger model itself often lacks sufficient language competence in the target LRL. As a result, when the larger model dominates the logit arithmetic, as implicitly assumed in Proxy Tuning (Liu et al., 2024; Zhao et al., 2024; Zhang et al., 2025c), its weak LRL representations can overwhelm the contribution of the smaller CPT model, limiting effective transfer and even destroying basic LRL generation ability (see an example in Appendix B.4). This observation highlights an important issue in logit-level fusion: different sources of model capability are not always equal and need more careful balancing.

To address these limitations, we propose TRIMIX, a test-time logit fusion framework integrating (i) language-specific competence acquired through CPT on LRLs, (ii) task competence learned from HRLs, and (iii) the scaling benefits of large models, as shown in Figure 1. TRIMIX requires CPT only on a small model using raw LRL text, without the need of task-level annotation. At inference time, it leverages an off-the-shelf large instruction-tuned model to benefit from scaling effects. Unlike prior Proxy Tuning approaches that treat the large model as the dominant component by default, TRIMIX explores adaptive weighting strategies to dynamically balance these different sources of capability.

We validate our framework on four LLM families across eight LRLs. TRIMIX, particularly when combined with the perplexity-guided weighting strategy, consistently outperforms the single-model baselines as well as Proxy Tuning. For instance, continually pretraining Qwen2.5-1.5B on LRL corpora and fusing it with a 14B instruction-tuned model yields an average relative improvement of approximately 5% over the 14B model.

To better understand the mechanisms underlying the effectiveness of TRIMIX, we analyze the weights assigned during logit fusion. Under the empirical upper bound of TRIMIX, we find that substantially larger weights are assigned to the small CPT model than to the large instruction-tuned model, in sharp contrast to Proxy Tuning, which typically prioritizes the larger model. Our perplexity-guided strategy for hyperparameter selection closely approximates the behavior of the upper bound setting, explaining its strong empirical performance. Furthermore, we show that divergence from the base model offers a plausible explanation for when language-specific competence should be emphasized.

Our contributions are summarized as follows: (1) We propose TRIMIX, an efficient test-time logit fusion framework for LRL, which dynamically integrates language-specific competence and task competence while leveraging the scaling benefits of large models. (2) We validate TRIMIX across three LLM families, covering multiple small–large model scale pairs and eight LRLs, demonstrating its flexibility and strong generalizability. (3) We challenge the large-model-dominant assumption underlying Proxy Tuning, showing the importance of prioritizing LRL competence in logit fusion.[1]

2 Method

We propose TRIMIX, a logit fusion framework designed to address the scarcity of annotated task data in LRLs while minimizing the cost of scaling model sizes. As illustrated in Figure 2, based on the logit arithmetic of three models, TRIMIX explicitly disentangles and recombines three sources of benefit: (i) language-specific competence acquired via CPT on LRL, (ii) task-solving capabilities transferred from HRLs, and (iii) general capability gains driven by scaling effects. To effectively integrate these different signals, we introduce an adaptive fusion mechanism that balances their contributions at inference time.

2.1 Preliminary

Logit Let $L \in \mathbb{R}^{|V|}$ denote the logit vector (the unnormalized token distribution) produced by a language model before the softmax operation, where $|V|$ is the vocabulary size of the model. Different models (or variants thereof) induce distinct logit distributions reflecting their abilities obtained through specific training data. We follow prior research, such as Proxy Tuning (Liu et al., 2024), that demonstrates that linearly combining logits from different models can effectively synthesize capabilities without further training.

Model Variants Our framework leverages open-source LLMs trained primarily on HRLs. We define three key variants:

- **Base (base)**: A foundation model pretrained on massive HRL corpora, exhibiting general reasoning and language modeling capabilities.
- **Instructed (ins)**: An instruction-tuned variant obtained by fine-tuning the base model on

[1] Our code are publicly available at https://github.com/luciusssss/TriMix.

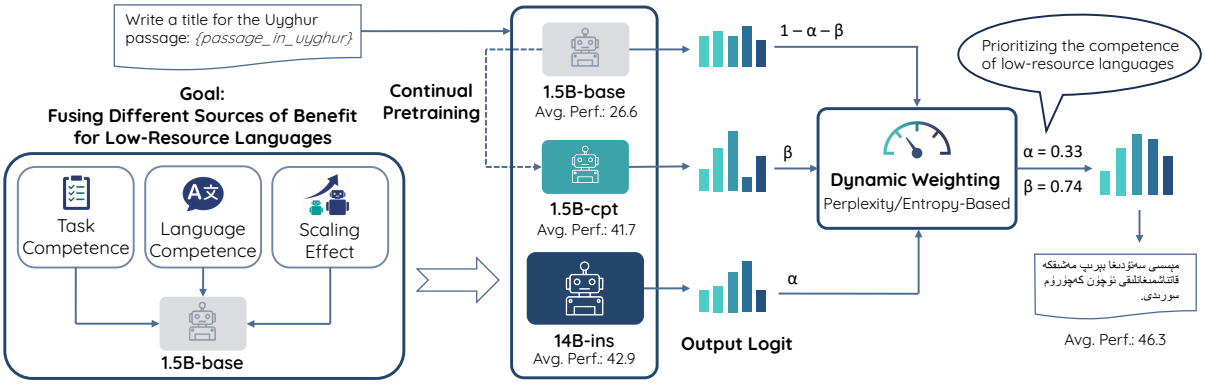

Figure 2: The framework of TRIMIX. Given a task prompt for an LRL, TRIMIX dynamically fuses the logits of three models to integrate language competence, task competence, and scaling benefits.

HRL instruction-following data.

- **Continually Pretrained (cpt)**: A variant obtained by performing CPT on the base model using LRL corpora to enhance target language competence.

To minimize computational overhead, we perform CPT solely on a small model variant (small-cpt) and transfer this competence to a larger model variant (large-ins) via logit fusion.

2.2 Tri-Source Fusion

We formulate the fusion objective as a linear decomposition of capabilities. We posit that the ideal logit L can be constructed by augmenting a baseline small model with specific benefit vectors:

$$L = L_{\text{small-base}} + \alpha\delta_{\text{T}} + \beta\delta_{\text{L}} + \gamma\delta_{\text{S}}, \quad (1)$$

where δ_{T}, δ_{L}, and δ_{S} represent the benefit vectors for task solving, LRL modeling, and scaling, respectively.

Task Solving Vector (δ_{T}) We isolate the task-solving capability by contrasting the large instruction-tuned model with its base counterpart. We utilize the large model here instead of the small one, as larger models typically exhibit stronger learning capacity under the same amount of data (Kaplan et al., 2020):

$$\delta_{\text{T}} = L_{\text{large-ins}} - L_{\text{large-base}}. \quad (2)$$

Language Modeling Vector (δ_{L}) Due to computational constraints, we assume that CPT is feasible only for the smaller model. Thus, we define the LRL competence vector as:

$$\delta_{\text{L}} = L_{\text{small-cpt}} - L_{\text{small-base}}. \quad (3)$$

Scaling Effect Vector (δ_{S}) To capture the benefits of scaling, we compute the difference between the large and small base models. One could subtract either the base versions or the instructed versions between the large and small models. We choose the base models to avoid potential confounding effects introduced by instruction tuning:

$$\delta_{\text{S}} = L_{\text{large-base}} - L_{\text{small-base}}. \quad (4)$$

To streamline the inference process and reduce memory overhead, we strategically set the scaling coefficient $\gamma = \alpha$. This constraint allows the $L_{\text{large-base}}$ terms in Eq. 2 and Eq. 4 to cancel out, eliminating the need to load the large base model into memory. As a result, both the task-solving capability and the scaling effect are injected through a single large instruction-tuned model[2]. Substituting the definitions into Eq. 1 yields our final formulation:

$$\begin{aligned} L = \alpha L_{\text{large-ins}} + \beta L_{\text{small-cpt}} \\ + (1 - \alpha - \beta)\, L_{\text{small-base}}. \end{aligned} \quad (5)$$

2.3 Dynamic Weighting

In the final formulation (Eq. 5), two hyperparameters must be determined. In LRL settings, we typically lack sufficient annotated data to perform an extensive hyperparameter search. Consequently, we propose two heuristics to determine these weights dynamically at inference time.

Perplexity-Guided Selection (PPL) The perplexity of the input prompt serves as a proxy for

[2]This constraint trades modeling flexibility for efficiency: while allowing $\gamma \neq \alpha$ might yield better performance, we adopt $\gamma = \alpha$ as a practical approximation and leave unconstrained fusion to future work.

Family	small-cpt	large-ins	Languages
Qwen2.5	1.5B/3B/7B	3B/7B/14B	bod, uig, kaz, mvf
Llama2	7B	13B	bod, uig, mvf, tam, tel, ory, ben
Llama3.2	1B	3B	bod, uig, kaz
Gemma3	4B	12B	bod, uig, kaz, mvf

Table 1: The model scales and evaluated languages of each model family.

how well the fused model captures the input distribution (Mavromatis et al., 2024; Xu et al., 2025). We select the (α, β) pair that minimizes the perplexity of the prompt, which consists of in-context learning examples and the input of the current test instance.

Entropy-Guided Selection (ENT) Alternatively, we utilize predictive confidence as a selection metric. Inspired by previous works (Garces Arias et al., 2024; Jin et al., 2024), we calculate the entropy of the next-token distribution. We select the hyperparameters that minimize the entropy of the first generated token, prioritizing configurations where the model exhibits high certainty.

3 Experiments

3.1 Experimental Setups

Models We primarily evaluate our framework using the different scales of models from **Qwen2.5** family (Yang et al., 2024). To assess the generalizability across architectures, we also conduct experiments with the **Llama2** (Touvron et al., 2023), **Llama3.2** (Grattafiori et al., 2024), and **Gemma3** (Team et al., 2025) series, which exhibit different levels of multilingual abilities. We choose these model families because they offer different range of model sizes, which is required for our experiments. For Llama2, we directly use the CPT checkpoints from Tao et al. (2024); for other model series, we continually pretrain the base models as described in Appendix A.2.

Languages We focus on eight LRLs spanning diverse linguistic families and scripts. See their linguistic details in Appendix A.1. We mainly evaluate on four minority languages in China, including Tibetan (bod), Uyghur (uig), Kazakh (kaz, Arabic script), and Mongolian (mvf, traditional script). For Llama2, we additionally evaluate on four Indian languages using the CPT checkpoints from Tao et al. (2024), including Tamil (tam), Telugu (tel), Odia (ory), and Bengali (ben). In Table 1, we summarize the model scales and evaluated languages[3].

Evaluation Datasets For the four minority languages in China, we adopt the **MiLiC-Eval** benchmark (Zhang et al., 2025b). We group its seven tasks into three categories: **(1) Multi-Choice (MC):** topic classification, response selection, and reading comprehension; **(2) Generation in English (ENG-G):** LRL-to-English translation and mathematical reasoning (with English Chain-of-Thought); **(3) Generation in LRL (LRL-G):** title generation and English-to-LRL translation. For the Indian languages, following Tao et al. (2024), we evaluate on the **Belebele** reading comprehension dataset (Bandarkar et al., 2024) and the **SIB-200** topic classification dataset (Adelani et al., 2024).

Baseline Methods We compare TriMix against established methods for capability transfer and logit manipulation.

Model Merging (Tao et al., 2024; Huang et al., 2024a; Akiba et al., 2025) combines homogeneous models with complementary capabilities by merging their parameters. In our setting, we merge a small-cpt model and a small-ins model to transfer task-solving capabilities learned from HRLs to LRLs. Following Tao et al. (2024), we adopt the widely used TIES algorithm (Yadav et al., 2023) for merging.

Contrastive Decoding (Li et al., 2023) amplifies the signal of a *strong* model by subtracting the logits of a *weak* model. In the LRL setting, we treat small-cpt as the strong model and small-base as the weak model to enhance language competence:

$$L = L_{\text{small-cpt}} + \beta(L_{\text{small-cpt}} - L_{\text{small-base}}). \quad (6)$$

Proxy Tuning (Liu et al., 2024) attempts to transfer the knowledge learned in a smaller model to a larger one by logit arithmetic. It constitutes a special case of our TriMix formulation where the instruction weight is fixed at $\alpha = 1$. The decoding objective becomes:

$$L = L_{\text{large-ins}} + \beta(L_{\text{small-cpt}} - L_{\text{small-base}}). \quad (7)$$

Liu et al. (2024) originally set β to 1 for simplicity, which performs well in practice. Consistent with

[3]For Llama2, the publicly available CPT checkpoints from Tao et al. (2024) do not cover Kazakh. For Llama3.2, we observe no noticeable improvement after CPT on Mongolian, likely due to inadequate tokenizer support and the limited capacity of the smaller model; therefore, we exclude this language from our experiments.

Method	#Param Train	#Param Test	MC	ENG-G	LRL-G	bod	uig	kaz	mvf	Average
small = 1.5B, large = 1.5B										
Qwen2.5-1.5B-base†	0B	1.5B	40.2	10.8	11.5	22.3	26.6	25.5	20.0	23.6
Qwen2.5-1.5B-ins†	0B	1.5B	36.4	10.4	10.5	19.8	24.3	23.4	18.9	21.6
Qwen2.5-1.5B-cpt	1.5B	1.5B	**48.3**	17.6	17.3	25.7	41.7	**33.8**	**21.5**	30.7
Contrastive Decoding	1.5B	3B	45.3	15.7	15.2	25.0	37.7	32.1	18.1	28.2 (-8.1%)
Model Merging	1.5B	1.5B	48.0	**18.6**	**17.5**	**27.0**	**41.8**	**33.8**	21.0	**30.9** (+0.7%)
small = 1.5B, large = 3B										
Qwen2.5-3B-ins†	0B	3B	42.4	12.2	10.8	24.2	30.4	23.2	21.2	24.8
Proxy Tuning	1.5B	6B	45.4	14.1	14.1	23.5	40.0	29.4	21.2	28.5 (-7.2%)
TRIMIX (ENT)	1.5B	6B	45.0	13.7	14.8	24.1	33.8	29.3	**22.5**	27.4 (-10.7%)
TRIMIX (PPL)	1.5B	6B	**48.7**	**19.5**	**16.3**	**25.6**	**41.4**	**36.4**	21.1	**31.1** (+1.3%)
TRIMIX *(Upper Bound)*	*1.5B*	*6B*	*52.4*	*21.3*	*17.6*	*27.9*	*43.8*	*37.4*	*25.3*	*33.6* (+9.4%)
small = 1.5B, large = 7B										
Qwen2.5-7B-ins†	0B	7B	49.7	**20.0**	12.5	**28.2**	40.6	30.5	23.0	30.6
Proxy Tuning	1.5B	10B	50.5	16.3	13.3	26.3	38.6	33.7	21.6	30.0 (-2.3%)
TRIMIX (ENT)	1.5B	10B	51.2	16.1	**16.4**	26.7	40.5	34.8	22.9	31.2 (+1.6%)
TRIMIX (PPL)	1.5B	10B	**53.4**	19.8	15.7	27.7	**42.7**	**38.1**	**23.6**	**33.0** (+7.5%)
small = 1.5B, large = 14B										
Qwen2.5-14B-ins†	0B	14B	57.1	**21.0**	13.8	**35.7**	42.9	34.6	24.6	34.4
Proxy Tuning	1.5B	17B	57.7	15.4	16.8	33.1	40.3	37.3	25.0	33.9 (-1.5%)
TRIMIX (ENT)	1.5B	17B	57.8	15.6	**17.0**	33.3	41.3	35.8	25.8	34.1 (-0.9%)
TRIMIX (PPL)	1.5B	17B	**59.5**	20.5	16.8	33.6	**46.3**	**38.6**	**26.0**	**36.1** (+4.9%)

Table 2: Evaluation results on Qwen2.5 models with different model size combinations. † denotes off-the-shelf single models. **#Param Train** and **#Param Test** denote the number of parameters updated during CPT and used during inference, respectively. **Bold** indicates the best result, and underlined indicates the second best. The numbers in parentheses in the **Average** column represent relative improvements over the best single-model baselines.

this choice, our experiments, such as the hyperparameter analysis in Figure 4, indicate that $\beta = 1$ generally yields better performance than smaller values when $\alpha = 1$.

Upper Bound: To estimate the performance ceiling of TRIMIX, we conduct an exhaustive grid search over α and β in Eq. 5 and report the optimal achievable performance for each task. Due to the substantial computational cost of evaluating 49 hyperparameter pairs across 28 tasks, we restrict this oracle search to the setting of Qwen2.5 1.5B-small + 3B-ins.

Implementation Details For hyperparameter selection in Sec. 2.3, we perform an efficient grid search over the discrete set $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$ for both α and β [4]. We estimate a set of optimal fusion hyperparameters for each task using a subsample of 50 examples. See Appendix A for implementation details of pretraining and inference.

[4]As discussed in Appendix B.1, we find that model performance degrades when the scales of logits from different models differ substantially; therefore, we restrict the search to this range to avoid excessive imbalance during fusion.

3.2 Main Results

In Table 2, we report the average performance of different methods on the Qwen2.5 family. See the detailed results on individual tasks in Appendix B.2.

Superiority of Tri-Source Fusion In the setting that combines 1.5B models with a 3B model, TRIMIX (PPL) outperforms single-model baselines. It also surpasses other multi-model collaboration methods, including model merging and contrastive decoding, when using the same CPT model size. These results highlight the effectiveness of our approach in leveraging the scaling benefits of larger models.

Scalability When scaling the instruction-following model from 3B to 14B (with the CPT model fixed at 1.5B), TRIMIX (PPL) yields consistent performance gains, achieving a +1.7% absolute improvement (+4.9% relative) over the strong 14B-ins baseline. On the other hand, scaling the CPT model, as shown in Figure 3, leads to steady improvements when combined with larger models of various sizes.

Model	bod	uig	mvf	tam	tel	ory	ben	Average
7B-base	17.0	18.3	12.9	26.4	21.8	19.3	31.7	21.1
7B-cpt	27.6	26.8	12.6	41.1	44.5	36.3	46.4	33.6
13B-ins	22.1	22.7	**16.4**	25.7	19.0	22.7	34.0	23.2
Proxy Tuning	25.7	26.8	13.3	33.2	39.6	33.3	47.3	31.3 (-6.8%)
TRIMIX (PPL)	**30.4**	**33.2**	13.8	**46.1**	**50.8**	**43.2**	**53.6**	**38.7** (+15.2%)

Table 3: Evaluation results on Llama2 models.

Model	bod	uig	kaz	mvf	Average
4B-base	24.2	32.0	24.7	17.2	24.5
4B-cpt	35.7	36.0	33.2	19.1	31.0
12B-ins	49.7	**57.6**	50.8	24.1	45.6
Proxy Tuning	49.6	54.4	48.5	24.6	44.3 (-2.9%)
TRIMIX (PPL)	**54.8**	55.9	**51.8**	**29.4**	**48.0** (+5.3%)

Table 4: Evaluation results on Gemma3 models.

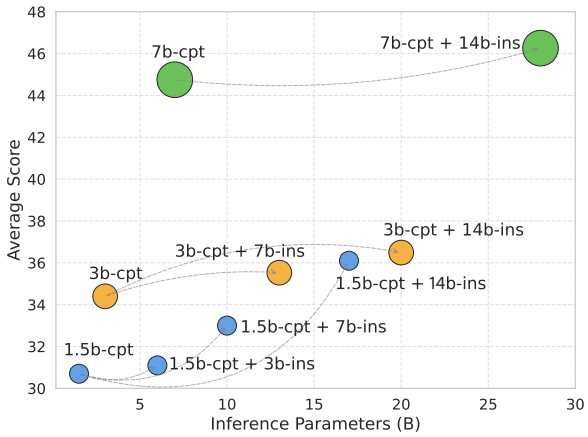

Figure 3: Performance of TRIMIX (PPL) across different combinations of model sizes. Scores are reported on MiLiC-Eval and averaged over four languages. Circle sizes represent the scale of training parameters.

In practice, LRL researchers can start with the largest CPT model they can feasibly train and apply TRIMIX in combination with a larger instruction-tuned model to achieve stable performance gains. In Appendix B.3, we additionally analyze the effectiveness of TRIMIX when the CPT model undergoes only partial training due to computational constraints.

Weighting Strategies Although Proxy Tuning is designed to exploit the capacity of larger models, it consistently underperforms in the LRL settings, often yielding worse results than even the standalone CPT model. We will further analyze the underlying reason in Section 4.

Among our two unsupervised selection strategies, Perplexity-Guided (PPL) selection generally outperforms Entropy-Guided (ENT) selection. We attribute this advantage to the fact that perplexity is computed over the entire input prompt, providing a global contextual signal, whereas entropy depends only on the next-token distribution. Still, a gap remains between our best heuristic and the Upper Bound, indicating that more advanced selection mechanisms could further improve performance.

3.3 Generalizability

To validate the generalizability of TRIMIX, we extend our evaluation to three additional LLM families (Gemma3, Llama2 and Llama3.2) and on four additional low-resource Indian languages.

Table 4 reports the results on the Gemma3 family. Gemma3-12B-ins is a particularly strong baseline, outperforming all Qwen2.5 configurations of logit fusion. Despite this high starting point, TRIMIX (combining 4B-cpt and 12B-ins) yields an average relative improvement of +5.3% over the best single model. This indicates that our method is still effective even when the backbone model is already capable in the target LRL.

We further test on Llama2 (7B-cpt + 13B-ins) covering four additional Indian languages, as shown in Table 3. TRIMIX demonstrates strong gains on Indian languages, improving over the 7B-cpt baseline by large margins (e.g., +5.1 on Tamil, +7.2 on Bengali). This further confirms that our framework generalizes effectively across language families. We report the results on Llama3.2 in Appendix B.2, where TRIMIX outperforms the single-model baseline by 8.7% relatively.

We observe that TRIMIX gains are occasionally marginal or negative for Tibetan (bod) and Mongolian (mvf) in certain configurations. We hypothesize this stems from the extremely low tokenizer fertility for these scripts in English-centric models. As noted by Zhang et al. (2025b), Qwen2.5's tokenization efficiency for Tibetan is approximately $10\times$ lower than for English. Future work will investigate vocabulary expansion techniques to mitigate this bottleneck.

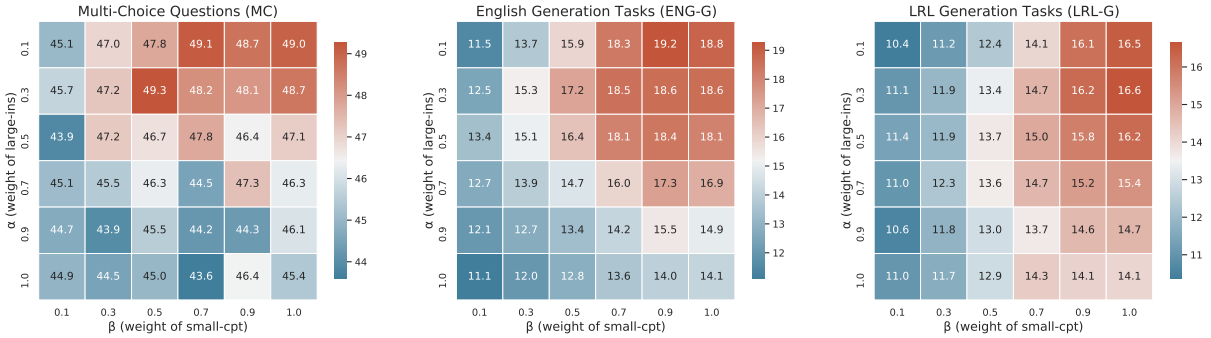

Figure 4: The average score across all the tasks on MiLiC-Eval for each pair of hyperparameters in TRIMIX.

4 Discussion

To better understand the reason behind the effectiveness of TRIMIX, we analyze the weights assigned to each model under different hyperparameter selection strategies, and interpret the weighting behaviors through the lens of model divergence from the base model.

4.1 Analysis of Hyperparameter Selection

To understand the mechanics of TRIMIX, we conduct a grid search of the fusion weights (α, β) for the Qwen2.5 1.5B-cpt + 3B-ins setting. Figure 4 visualizes the average performance across all four LRLs on MiLiC-Eval for different hyperparameter combinations. We observe a consistent pattern across tasks: the optimal region is situated in the high β, low α quadrant, generally satisfying the condition $\beta > \alpha$. This indicates that significantly greater weight must be assigned to the smaller LRL-adapted CPT model ($L_{\text{small-cpt}}$) than to the large instruction-tuned model ($L_{\text{large-ins}}$). This empirical finding directly contradicts the core assumption of Proxy Tuning (where $\alpha = 1$ is by default), providing the potential explanation for its sub-optimal performance in LRL scenarios.

Table 5 reports the mean selected hyperparameters of different strategies, under the Qwen2.5 1.5B-cpt + 3B-ins setting. The perplexity-guided strategy yields a mean configuration (α = 0.11, β = 0.91) that closely mirrors the distribution of the Upper Bound (α = 0.33, β = 0.74). In contrast, the entropy-guided strategy (α = 0.95, β = 0.61) assigns excessively high weight to the large instructed model, similar to the ineffective Proxy Tuning approach. To summarize, the alignment between the PPL heuristic and the upper bound configuration explains its consistent superiority in the main results (Table 2).

Model	α	β
Upper Bound	0.33±0.26	0.74±0.29
Proxy Tuning	1.00±0.00	1.00±0.00
TRIMIX (ENT)	0.95±0.14	0.61±0.40
TRIMIX (PPL)	0.11±0.05	0.91±0.06

Table 5: Mean and standard deviation of hyperparameters (α, β) selected by different fusion strategies.

4.2 Explaining the Balance of Competence

Previous studies (Jacot et al., 2018; Korbak et al., 2022) suggest that changes in a model's output distribution are closely correlated with changes in its parameters. Accordingly, a large divergence between the logits before and after specialized learning indicates that the model has undergone sufficient specialization. Building on this presumption, we hypothesize that a specialized model should receive a larger fusion weight when it remains closer to the common base model. When the component models of logit fusion exhibit asymmetric divergence, their fusion weights should be recalibrated accordingly to balance their specialized contributions during fusion.

Setup To validate our hypothesis, we conduct an analysis in two domains: (i) the code domain studied in the original Proxy Tuning work, and (ii) LRLs. We quantify how specialized models differ from the base model by computing the KL divergence between the predicted token distributions of the first generated token under identical inputs.

For the code domain, we consider three models: Qwen2.5-1.5B-base, Qwen2.5-1.5B-coder (Hui et al., 2024), and Qwen2.5-3B-ins. We compute the KL divergence between 1.5B-coder and 1.5B-base to capture the effect of code-domain adaptation, and between 3B-ins and 1.5B-base to measure the effect of instruction tuning combined with scaling. The measurements are conducted on the

Task	3B-ins	1.5B-coder
HumanEval	0.140	0.142
Task	**3B-ins**	**1.5B-cpt**
MiLiC-Eval (bod)	1.387	0.972
MiLiC-Eval (uig)	2.124	1.080
MiLiC-Eval (kaz)	2.233	1.077
MiLiC-Eval (mvf)	1.859	1.823

Table 6: KL divergence of various model variants relative to Qwen2.5-1.5B-base. Scores are calculated using the predicted distributions of the first generated token.

HumanEval code completion benchmark (Chen et al., 2021), following the original work. For LRLs, we use models from the same Qwen2.5 family: Qwen2.5-1.5B-base, Qwen2.5-1.5B-cpt, and Qwen2.5-3B-ins. We compute the KL divergence between 1.5B-cpt and 1.5B-base to measure the effect of continual pretraining on LRL data. Evaluation is conducted on MiLiC-Eval, averaging the results over 7 tasks.

Results Table 6 reports the KL divergence between different specialized models and a shared base model. In the code domain, the KL divergence between 1.5B-coder and 1.5B-base is comparable to that between 3B-ins and 1.5B-base. This suggests that the code CPT trained on over 5T tokens of code induces parameter and distributional changes of a similar magnitude to those introduced by instruction tuning combined with scaling. Under this condition, assigning equal fusion weights (i.e., $\alpha = \beta = 1$) is reasonable to achieve strong performance on code-related tasks, which explains the empirical success of Proxy Tuning in this domain.

In contrast, a markedly different pattern emerges in LRL settings. The KL divergence between 3B-ins and 1.5B-base is generally larger than that between the 1.5B-cpt and 1.5B-base. This indicates that instruction tuning and scaling introduce substantially stronger distributional shifts than LRL CPT. As a result, Proxy Tuning tends to overemphasize the large instruction-tuned model, thereby suppressing the LRL competence learned during CPT. Notably, LRL CPT typically involves fewer than 1B tokens, and the resulting language competence is far from saturated, suggesting that it should receive a larger fusion weight. These observations motivate rebalanced fusion in LRL scenarios, where the LRL component must be up-weighted ($\beta > \alpha$) to compensate for its lower intrinsic divergence from the base model.

Setting	Score
TRIMIX	38.1
w/o Task Solving Vector δ_T	37.2 (-0.9)
w/o Language Modeling Vector δ_L	30.5 (-7.6)
w/o Scaling Effect Vector δ_S	33.9 (-4.2)

Table 7: Ablation study of the three benefit vectors in TRIMIX.

Overall, these findings provide empirical support for our hypothesis that optimal fusion weights are closely tied to each component's divergence from the base model, and that different domains exhibit fundamentally different divergence structures, motivating the adaptive weighting strategy of TRIMIX.

4.3 Ablation of Benefit Sources

To understand how each source of benefit in TRIMIX affects the overall performance, we conduct an ablation study of the three benefit vectors in Eq. 1. In Table 7, we report the performance of Qwen2.5 1.5B-cpt + 7B-ins combination on the Kazakh tasks of MiLiC-Eval, using perplexity-based hyperparameter selection. The largest degradation occurs when removing the language modeling vector (δ_L), indicating that language modeling ability is the dominant contributor. The scaling effect (δ_S) is the second most important, while the task-solving vector (δ_T) provides smaller but consistent gains. These findings align with our observation that upweighting language ability (high β) is critical in LRL settings.

5 Related Works

Language Adaptation Existing approaches for adapting LLMs to LRLs can be broadly categorized into two paradigms: Parameter-level approaches rely on continual pretraining (Yong et al., 2023; Fujii et al., 2024; Aggarwal et al., 2025; Li et al., 2025a) or fine-tuning (Su et al., 2024; Singh et al., 2024; Shaham et al., 2024) on LRL data to improve language competence; prompt-level approaches teach LLMs new languages by injecting external linguistic resources, such as dictionaries (Zhang et al., 2024a; Li et al., 2025b) and grammar books (Tanzer et al., 2024; Hus and Anastasopoulos, 2024; Zhang et al., 2025a; Pei et al., 2025) into the input context.

Beyond adapting a single model, recent work has explored multi-model collaboration to combine

capabilities for LRLs. Parameter-level techniques include model merging (Tao et al., 2024; Huang et al., 2024a; Cao et al., 2025; Yamaguchi et al., 2025), layer swapping (Bandarkar et al., 2025), and model stacking (Huang et al., 2024b; Schmidt et al., 2024; Su et al., 2025). At the prompt level, post-editing methods (Cheng et al., 2024; Li et al., 2024; Deoghare et al., 2024) are widely used.

In contrast to these approaches, we explore an orthogonal yet underexplored direction for LRL adaptation: logit-level fusion, which is computationally efficient than parameter updating and provides finer-grained control over model behavior than prompt-level methods.

Logit Fusion Logit fusion aims to combine token-level output distributions from multiple models during inference, addressing the challenges in hallucinations (Shi et al., 2024; Kim et al., 2024), reasoning (O'Brien and Lewis, 2023; Tao et al., 2025; Zhang et al., 2025c), and safety (Zhong et al., 2024; Fu et al., 2025). One line of works amplifies desirable behaviors of stronger models by contrasting them with weaker ones, typically through contrastive decoding (Li et al., 2023; Chuang et al., 2024; Yu et al., 2025). The other line combines complementary capabilities across models. This includes transferring specialized abilities learned by smaller models to larger ones (Liu et al., 2024; Mitchell et al., 2024), as well as directly ensembling multiple models (Mavromatis et al., 2024). To guide fusion, prior work has explored confidence- or uncertainty-based signals (Garces Arias et al., 2024; Jin et al., 2024; Lee et al., 2025), divergence-based criteria (Fan et al., 2024), or learned routers (Shen et al., 2024).

In multilingual contexts, preliminary efforts have applied logit fusion to mitigate hallucinations in machine translation (Sennrich et al., 2024; Waldendorf et al., 2024) or to improve multilingual mathematical reasoning (Zhu et al., 2024). However, they are primarily tailored for high-resource languages and typically do not account for scaling effects. In contrast, we systematically analyze the limitations of existing logit fusion strategies in LRL settings and propose a refined framework that balances various sources of benefits.

6 Conclusion

We propose TRIMIX, a logit fusion framework for LRLs that balances language competence, task competence, and scaling benefits while requiring continual pretraining only on a small model. Experiments across four LLM families and eight languages demonstrate its effectiveness and generalizability. Our analysis shows that prioritizing language-specific competence is crucial for LRLs, challenging the large-model-dominant assumption in prior work. We hope that TRIMIX offers a practical path for LRL adaptation under limited data and computational budgets.

Limitations

Effect of Vocabulary Expansion TRIMIX does not explicitly examine the effect of vocabulary expansion, nor does it address potential vocabulary or tokenization misalignment across component models. In our setting, all models share a largely compatible tokenizer, which allows for direct logit-level fusion without additional alignment mechanisms. However, for languages with underrepresented scripts, it is often suggested to expand the vocabulary before CPT to improve encoding efficiency. This might result in mismatched vocabularies between component models during logit fusion, requiring additional preprocessing steps such as vocabulary mapping or re-tokenization, which we leave for future work.

Applicability to Closed-Source LLMs TRIMIX requires direct access to the output logits of all component models in order to perform logit-level fusion. This assumption restricts its applicability to open-source models or locally deployed systems. For closed-source or API-based models, where only final decoded outputs are available, TRIMIX cannot be directly applied. Exploring proxy signals or alternative fusion strategies that do not rely on raw logits remains an open challenge.

Inference Overhead While TRIMIX significantly reduces the computational cost associated with continual pretraining over large models, it introduces additional overhead at inference time. Specifically, inference requires running multiple component models in parallel and performing logit fusion at each generation step. This increases latency and memory consumption compared to single-model inference. This overhead can be partially mitigated through techniques such as model quantization. For instance, with INT8 quantization, the quantized TRIMIX configuration still significantly outperforms the strongest single-model baseline, with less than a 0.5% performance drop

compared to the non-quantized configuration. Future work could investigate more sophisticated techniques like model compression, selective activation, or distillation-based variants of TRIMIX to mitigate this overhead.

Potential Risks Although TRIMIX improves performance on many LRL benchmarks, it may still produce incorrect or nonsensical outputs in these languages and should therefore be used with caution.

Acknowledgements

This work is supported in part by Beijing Natural Science Foundation (L253001) and Natural Science Foundation of China (92570207). We thank the anonymous reviewers for their valuable feedback. For any correspondence, please contact Yansong Feng.

References

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. 2024. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta. Association for Computational Linguistics.

Divyanshu Aggarwal, Ashutosh Sathe, and Sunayana Sitaram. 2025. Improving cross lingual transfer by pretraining with active forgetting. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2378, Suzhou, China. Association for Computational Linguistics.

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. 2025. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7(2):195–204.

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

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

Sheng Cao, Mingrui Wu, Karthik Prasad, Yuandong Tian, and Zechun Liu. 2025. Paramδ for direct mixing: Post-train large language model at zero cost. In *The Thirteenth International Conference on Learning Representations*.

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.

Xin Cheng, Xun Wang, Tao Ge, Si-Qing Chen, Furu Wei, Dongyan Zhao, and Rui Yan. 2024. SCALE: Synergized collaboration of asymmetric language translation engines. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15903–15918, Bangkok, Thailand. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Sourabh Deoghare, Diptesh Kanojia, and Pushpak Bhattacharyya. 2024. Together we can: Multilingual automatic post-editing for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10800–10812, Miami, Florida, USA. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chenghao Fan, Zhenyi Lu, Wei Wei, Jie Tian, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024. On giant's shoulders: Effortless weak to strong by dynamic logits fusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Tingchen Fu, Yupeng Hou, Julian McAuley, and Rui Yan. 2025. Unlocking decoding-time controllability: Gradient-free multi-objective alignment with contrastive prompts. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 366–384, Albuquerque, New Mexico. Association for Computational Linguistics.

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 *First Conference on Language Modeling*.

Esteban Garces Arias, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. 2024. Adaptive contrastive search: Uncertainty-guided decoding for open-ended text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15060–15080, Miami, Florida, USA. Association for Computational Linguistics.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Andrés Domínguez Hernández, Diana Mosquera, and Francisco Gallegos. 2025. Lessons from the margins: Contextualizing, reimagining, and hacking generative ai in the global south. *Harvard Data Science Review*, 7(4).

Shih-Cheng Huang, Pin-Zu Li, Yu-chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tsai, and Hung-yi Lee. 2024a. 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.

Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. 2024b. Mindmerger: Efficiently boosting LLM reasoning in non-english languages. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Jonathan Hus and Antonios Anastasopoulos. 2024. Back to school: Translation using grammar books. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20207–20219, Miami, Florida, USA. Association for Computational Linguistics.

Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Jing Jin, Houfeng Wang, Hao Zhang, Xiaoguang Li, and Zhijiang Guo. 2024. DVD: Dynamic contrastive decoding for knowledge amplification in multi-document question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4624–4637, Miami, Florida, USA. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Zixuan Ke, Yifei Ming, and Shafiq Joty. 2025. Adaptation of large language models. In *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts)*, pages 30–37, Albuquerque, New Mexico. Association for Computational Linguistics.

Youna Kim, Hyuhng Joon Kim, Cheonbok Park, Choonghyun Park, Hyunsoo Cho, Junyeob Kim, Kang Min Yoo, Sang-goo Lee, and Taeuk Kim. 2024. Adaptive contrastive decoding in retrieval-augmented generation for handling noisy contexts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2421–2431, Miami, Florida, USA. Association for Computational Linguistics.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. 2022. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220.

Hakyung Lee, Subeen Park, Joowang Kim, Sungjun Lim, and Kyungwoo Song. 2025. Uncertainty-aware contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26376–26391, Vienna, Austria. Association for Computational Linguistics.

Chong Li, Yingzhuo Deng, Jiajun Zhang, and Chengqing Zong. 2025a. Group then scale: Dynamic mixture-of-experts multilingual language model. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1730–1754, Vienna, Austria. Association for Computational Linguistics.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Yue Li, Zhixue Zhao, and Carolina Scarton. 2025b. It's all about in-context learning! teaching extremely low-resource languages to LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29532–29547, Suzhou, China. Association for Computational Linguistics.

Zhuang Li, Levon Haroutunian, Raj Tumuluri, Philip Cohen, and Reza Haf. 2024. Improving cross-domain low-resource text generation through LLM post-editing: A programmer-interpreter approach. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 347–354, St. Julian's, Malta. Association for Computational Linguistics.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024. Tuning language models by proxy. In *First Conference on Language Modeling*.

Costas Mavromatis, Petros Karypis, and George Karypis. 2024. Pack of LLMs: Model fusion at test-time via perplexity optimization. In *First Conference on Language Modeling*.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. 2024. An emulator for fine-tuning large language models using small language models. In *The Twelfth International Conference on Learning Representations*.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*.

Renhao Pei, Yihong Liu, Peiqin Lin, François Yvon, and Hinrich Schuetze. 2025. Understanding in-context machine translation for low-resource languages: A case study on Manchu. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8767–8788, Vienna, Austria. Association for Computational Linguistics.

Fabian David Schmidt, Philipp Borchert, Ivan Vulić, and Goran Glavaš. 2024. Self-distillation for model stacking unlocks cross-lingual NLU in 200+ languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6724–6743, Miami, Florida, USA. Association for Computational Linguistics.

Rico Sennrich, Jannis Vamvas, and Alireza Mohammadshahi. 2024. Mitigating hallucinations and off-target machine translation with source-contrastive and language-contrastive decoding. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 21–33, St. Julian's, Malta. Association for Computational Linguistics.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2304–2317, Bangkok, Thailand. Association for Computational Linguistics.

Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. 2024. Learning to decode collaboratively with multiple language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12974–12990, Bangkok, Thailand. Association for Computational Linguistics.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

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

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

Tong Su, Xin Peng, Sarubi Thillainathan, David Guzmán, Surangika Ranathunga, and En-Shiun Lee.

2024. Unlocking parameter-efficient fine-tuning for low-resource language translation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4217–4225, Mexico City, Mexico. Association for Computational Linguistics.

Zeli Su, Ziyin Zhang, Guixian Xu, Jianing Liu, Xu Han, Ting Zhang, and Yushuang Dong. 2025. Multilingual encoder knows more than you realize: Shared weights pretraining for extremely low-resource languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18259–18270, Vienna, Austria. Association for Computational Linguistics.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. A benchmark for learning to translate a new language from one grammar book. In *The Twelfth International Conference on Learning Representations*.

Mingxu Tao, Jie Hu, Mingchuan Yang, Yunhuai Liu, Dongyan Zhao, and Yansong Feng. 2025. EpiCoDe: Boosting model performance beyond training with extrapolation and contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14874–14885, Vienna, Austria. Association for Computational Linguistics.

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.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Gorka Urbizu, Ander Corral, Xabier Saralegi, and Iñaki San Vicente. 2025. Sub-1B language models for low-resource languages: Training strategies and insights for Basque. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pages 519–530, Suzhuo, China. Association for Computational Linguistics.

Jonas Waldendorf, Barry Haddow, and Alexandra Birch. 2024. Contrastive decoding reduces hallucinations in large multilingual machine translation models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2539, St. Julian's, Malta. Association for Computational Linguistics.

Yangyifan Xu, Jianghao Chen, Junhong Wu, and Jiajun Zhang. 2025. Hit the sweet spot! span-level ensemble for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8314–8325, Abu Dhabi, UAE. Association for Computational Linguistics.

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

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*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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.

Byeongho Yu, Changhun Lee, Jun-gyu Jin, and Eunhyeok Park. 2025. PruneCD: Contrasting pruned self model to improve decoding factuality. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32450–32461, Suzhou, China. Association for Computational Linguistics.

Chen Zhang, Jiuheng Lin, Xiao Liu, Zekai Zhang, and Yansong Feng. 2025a. Read it in two steps: Translating extremely low-resource languages with code-augmented grammar books. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3977–3997, Vienna, Austria. Association for Computational Linguistics.

Chen Zhang, Xiao Liu, Jiuheng Lin, and Yansong Feng. 2024a. Teaching large language models an unseen language on the fly. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8783–8800, Bangkok, Thailand. Association for Computational Linguistics.

Chen Zhang, Mingxu Tao, Quzhe Huang, Jiuheng Lin, Zhibin Chen, and Yansong Feng. 2024b. MC^2: Towards transparent and culturally-aware NLP for minority languages in China. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

pages 8832–8850, Bangkok, Thailand. Association for Computational Linguistics.

Chen Zhang, Mingxu Tao, Zhiyuan Liao, and Yansong Feng. 2025b. MiLiC-eval: Benchmarking multilingual LLMs for China's minority languages. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11086–11102, Vienna, Austria. Association for Computational Linguistics.

Yunxiang Zhang, Muhammad Khalifa, Lechen Zhang, Xin Liu, Ayoung Lee, Xinliang Frederick Zhang, Farima Fatahi Bayat, and Lu Wang. 2025c. Logit arithmetic elicits long reasoning capabilities without training. *arXiv preprint arXiv:2510.09354*.

Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang, and William Yang Wang. 2024. Weak-to-strong jailbreaking on large language models. In *ICML 2024 Next Generation of AI Safety Workshop*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2024. ROSE doesn't do that: Boosting the safety of instruction-tuned large language models with reverse prompt contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13721–13736, Bangkok, Thailand. Association for Computational Linguistics.

Wenhao Zhu, Sizhe Liu, Shujian Huang, Shuaijie She, Chris Wendler, and Jiajun Chen. 2024. Multilingual contrastive decoding via language-agnostic layers skipping. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8775–8782, Miami, Florida, USA. Association for Computational Linguistics.

A Implementation Details

A.1 Selected Languages

Table 8 summarizes the linguistic details and speaker populations of the target languages, which collectively have over 475 million speakers but receive limited support from existing LLMs.

Name	Family	Script	Population
Tibetan (bod)	Sino-Tibetan	Tibetan	7M
Uyghur (uig)	Turkic	Arabic	12M
Kazakh (kaz)	Turkic	Arabic	1.6M
Mongolian (mvf)	Mongolic	Mongolian	6M
Tamil (tam)	Dravidian	Tamil	79M
Telugu (tel)	Dravidian	Telugu	96M
Odia (ory)	Indo-Euro.	Odia	35M
Bengali (ben)	Indo-Euro.	Bengali	240M

Table 8: Language families, writing systems, and populations of the LRLs in our study.

A.2 Continual Pretraining

Data For continual pretraining of Qwen2.5-1.5B, Gemma3-4B, Llama3.2-1B on the minority languages in China, we use the MC^2 corpus (Zhang et al., 2024b). To mitigate catastrophic forgetting of English capabilities, we additionally incorporate English data from C4 (Dodge et al., 2021), amounting to 20% of the size of the target-language corpus.

Hyperparameters We perform continual pretraining using the Megatron framework (Shoeybi et al., 2019). The models are trained for one epoch with a batch size of 1M tokens, a learning rate of 2×10^{-5}, and a warmup ratio of 0.01 on eight L40S GPUs.

A.3 Baselines

Model Merging For model merging, we use Arcee's MergeKit (Goddard et al., 2024). For the CPT model, the density is set to 1.0 and the weight to 0.2. For the instruction-tuned model, the density is set to 0.2 and the weight to 0.8.

Logit Arithmetic For Proxy Tuning, we use the official implementation provided by Liu et al. (2024)[5]. We adapt the code to support contrastive decoding in our experimental setup. For the contrsative decoding, we set β to 0.5 and set the plausibility threshold to 0.1, following the original paper.

A.4 Evaluation

Data In MiLiC-Eval, both topic classification and machine translation consist of two subsets. We use the *Passage* subset for the topic classification task and the *Dialogue* subset for the machine translation task. For title generation, since the inputs are long and evaluation is computationally expensive, we randomly sample 200 instances for evaluation. We report the test instance counts per language and corresponding metrics in Table 9. We use INT8 quantization to reduce GPU memory consumption during inference.

Our use of these existing artifacts is solely for evaluating multilingual models and is consistent with their intended purposes.

Hyperparameters We adopt a 5-shot in-context learning setting during evaluation. When the input exceeds the maximum context length of a model, we reduce the number of in-context examples accordingly. For each experimental setting, we evaluate up to 28 datasets (4 languages $\times$ 7 tasks). To improve efficiency, each setting is evaluated only

[5] https://github.com/alisawuffles/proxy-tuning

Dataset	Size	Metric
MiLiC-Eval		
Topic Classification	600	Accuracy
Response Selection	507	Accuracy
Reading Comprehension	250	Accuracy
Title Generation	200	ROUGE-L
English-to-LRL Translation	773	chrF++
LRL-to-English Translation	773	chrF++
Math Reasoning	250	Accuracy
Belebele	900	Accuracy
SIB-200	204	Accuracy

Table 9: Summary of evaluation datasets with test instance counts per language and corresponding metrics.

α \ β	**0.1**	**1**	**5**	**10**
0.1	52.5	54.5	3.0	3.5
1	50.5	54.0	13.0	6.0
5	48.0	48.5	48.0	18.5
10	48.5	48.0	49.5	36.0

Table 10: Accuracy (%) on the Kazakh reading comprehension task from MiLiC-Eval, using a larger range of hyperparameters, in the setting of Qwen2.5-1.5B-cpt + Qwen2.5-3B-ins.

once. To ensure fairness and comparability across settings, we use the same random seed throughout all experiments.

B Additional Results

B.1 Larger Hyperparameter Search Range

In our experiments, we restrict the hyperparameter search range to $[0, 1]$. When extending this range, we observe that model performance degrades substantially as the logit scales of different component models become increasingly mismatched. For example, in the setting of Qwen2.5 1.5B-cpt + 3B-ins, Table 10 reports the performance on the Kazakh reading comprehension task from MiLiC-Eval under different hyperparameter values $\{0.1, 1, 5, 10\}$. When either fusion weight becomes large, performance drops sharply, falling below 5% in extreme cases.

B.2 Full Evaluation Results

Llama2 In Table 12, we report the results of Llama2 on SIB-200 and Belebele in the four Indian languages.

Llama3.2 In Table 11, we report the evaluation results on the Llama3.2 series. TRIMIX improves over the strongest single model by +2.2 points (8.7% relative).

Model	bod	uig	kaz	Average
1B-base	16.2	17.3	15.5	16.3
1B-cpt	16.8	22.4	21.2	20.2
3B-ins	19.6	30.8	25.1	25.2
TRIMIX (PPL)	20.9	33.2	28.1	27.4 (+8.7%)

Table 11: Evaluation results on Llama3.2 models.

Qwen2.5 We report the full results of the Qwen2.5 series on the MiLiC-Eval benchmark: Table 15 for Tibetan, Table 16 for Uyghur, Table 17 for Kazakh, and Table 18 for Mongolian.

B.3 Fusion After Inadequate CPT

In practice, computational constraints often prevent full continual pretraining (CPT). We therefore examine whether TRIMIX remains effective when applied to partially trained models.

To simulate a limited-budget CPT setting, we conduct experiments with Qwen2.5-7B and evaluate performance on seven Kazakh tasks from MiLiC-Eval. As shown in Table 13, even after only one quarter of the full CPT steps, TRIMIX yields a relative improvement of 5.4%.

Importantly, TriMix acts as a multiplier rather than an alternative for CPT. Even partially trained models benefit when combined with a stronger instruction-tuned model. This demonstrates that practitioners can maximize limited CPT budgets by leveraging inference-time combination with a larger off-the-shelf model.

B.4 Case Study

Table 14 presents a qualitative case study comparing different logit fusion methods. For a prompt that requires generating a story title in Uyghur, Proxy Tuning produces invalid byte sequences that cannot be decoded into meaningful UTF-8 characters. This failure suggests that Proxy Tuning's large-model-dominant assumption allows weak LRL representations in the large model to overwhelm the contribution of the smaller continual-pretrained model. In contrast, TRIMIX well balances the contributions of different models and generates a plausible Uyghur title, demonstrating its effectiveness in LRL settings.

Model	SIB-200	Belebele	Avg.
Tamil			
7B-base	27.3	25.4	26.4
7B-cpt	53.5	28.6	41.1
13B-ins	26.3	25.0	25.6
Proxy Tuning	32.3	34.1	33.2
TRIMIX (PPL)	56.6	32.6	46.1
Telugu			
7B-base	18.2	25.3	21.8
7B-cpt	64.6	24.2	44.5
13B-ins	24.2	13.7	19.0
Proxy Tuning	50.5	28.7	39.6
TRIMIX (PPL)	71.7	29.9	50.8
Odia			
7B-base	18.2	20.3	19.3
7B-cpt	52.5	20.1	36.3
13B-ins	25.3	20.1	22.7
Proxy Tuning	41.4	25.1	33.3
TRIMIX (PPL)	60.6	25.7	43.2
Bengali			
7B-base	36.4	26.9	31.7
7B-cpt	66.7	26.1	46.4
13B-ins	40.4	27.6	34.0
Proxy Tuning	61.6	33.1	47.3
TRIMIX (PPL)	74.7	32.4	53.6

Table 12: Accuracy (%) of different methods on SIB-200 and Belebele, using the Llama2 models.

Model	Score
14B-ins	34.6
7B-cpt (1/8 steps)	38.1
7B-cpt (1/4 steps)	42.1
TRIMIX (1/8 steps)	41.3 (8.4%)
TRIMIX (1/4 steps)	44.2 (5.0%)

Table 13: Performance of TRIMIX using checkpoints of incomplete CPT. The numbers in parentheses represent relative improvements over the best single-model baseline.

Input: Please write a title for the following article in Uyghur:
{A news story about Messi in Uyghur.}

Gold:
مېسسى بۇرۇن پارىژ سانت گېرماندىن كەچۈرۈم سورىغاندەك دۆلىتىمىز مەستانىلىرىدىنمۇ كەچۈرۈم سورامدۇ؟
Will Messi apologize to our fans, just like he apologized to Paris Saint-Germain?

Proxy Tuning:
مېسسى ئالدىنقى ھەپتە ئارام ئالماي سەئ ��
Messi didn't rest last week [unreadable characters]

TriMix:
مېسسى ئۆزىنىڭ شەخسىي ئىجتىمائىي ئالاقە تورى ئارقىلىق ئۆزىنىڭ سەئۇدىغا بېرىپ كۆڭۈلۈنىڭ مەشىقىگە قاتناشمىغانلىقى توغرىسىدا، كۆڭۈب ۋە سەپداشلىرىدىن رەسمىي كەچۈرۈم سورىدى
Messi officially apologized to the club and his teammates via his personal social media regarding his absence from club training due to his trip to Saudi Arabia.

Table 14: Case study of different methods on the title generation task in Uyghur, under the setting of Llama2 7B-cpt + 13B-ins.

Model	Topic CLS	Read. Comp.	Resp. Sel.	Title Gen.	MT xx2en	MT en2xx	Math	Avg.
Qwen2.5-1.5B-base	28.9	43.8	35.1	21.6	8.4	8.8	9.3	22.3
Qwen2.5-1.5B-cpt	48.9	45.3	30.8	24.1	11.2	14.1	5.5	25.7
Qwen2.5-1.5B-ins	28.1	34.3	33.3	13.5	10.6	11.9	6.5	19.8
Contrastive Decoding	61.1	40.5	26.5	18.4	13.3	14.4	0.5	25.0
Model Merging	49.8	43.0	36.3	27.6	12.1	12.5	7.8	27.0
Qwen2.5-3B-ins	34.1	42.5	39.0	24.1	12.9	12.5	4.5	24.2
Proxy Tuning (1.5B-cpt + 3B-ins)	33.9	43.0	39.1	18.8	14.0	13.3	2.5	23.5
TRIMIX (ENT) (1.5B-cpt + 3B-ins)	38.9	43.0	37.8	19.9	11.9	14.4	2.5	24.1
TRIMIX (PPL) (1.5B-cpt + 3B-ins)	43.9	43.0	36.9	23.6	12.5	14.3	5.0	25.6
TRIMIX (Upper Bound) (1.5B-cpt + 3B-ins)	46.0	50.0	40.5	24.0	14.0	14.4	6.5	27.9
Qwen2.5-7B-ins	39.4	48.3	39.6	29.4	14.2	13.2	13.3	28.2
Proxy Tuning (1.5B-cpt + 7B-ins)	46.8	45.0	35.1	22.2	15.7	13.6	5.5	26.3
TRIMIX (ENT) (1.5B-cpt + 7B-ins)	45.4	44.5	37.1	24.4	15.7	14.3	5.5	26.7
TRIMIX (PPL) (1.5B-cpt + 7B-ins)	58.7	44.5	38.1	22.3	12.8	14.1	3.5	27.7
Qwen2.5-14B-ins	74.8	53.0	41.0	29.9	20.9	15.0	15.0	35.7
Proxy Tuning (1.5B-cpt + 14B-ins)	77.0	54.5	38.6	21.5	21.3	15.8	3.0	33.1
TRIMIX (ENT) (1.5B-cpt + 14B-ins)	74.0	54.5	40.1	21.5	21.3	16.4	5.0	33.3
TRIMIX (PPL) (1.5B-cpt + 14B-ins)	84.5	50.5	38.6	21.8	18.0	15.9	5.5	33.6

Table 15: Scores (%) of different methods on the **Tibetan** tasks of MiLiC-Eval, using the Qwen2.5 models. **Topic CLS** refers to topic classification. **Read. Comp.** refers to Reading Comprehension. **Resp. Sel.** refers to Response Selection. **Title Gen.** refers to Title Generation. **MT** refers to Machine Translation. xx2en denotes translation from LRLs to English. en2xx denotes translation from English to LRLs. **Math** refers to Math Reasoning.

Model	Topic CLS	Read. Comp.	Resp. Sel.	Title Gen.	MT xx2en	MT en2xx	Math	Avg.
Qwen2.5-1.5B-base	47.5	48.7	38.7	15.3	13.1	6.6	16.2	26.6
Qwen2.5-1.5B-cpt	81.3	65.3	44.8	16.8	37.2	20.5	25.7	41.7
Qwen2.5-1.5B-ins	39.0	44.2	36.4	14.4	14.1	9.4	12.3	24.3
Contrastive Decoding	72.2	59.0	40.8	15.6	37.2	19.9	19.5	37.7
Model Merging	83.7	61.5	46.0	18.4	38.4	20.9	24.0	41.8
Qwen2.5-3B-ins	71.4	38.2	39.6	17.7	19.9	10.6	15.5	30.4
Proxy Tuning (1.5B-cpt + 3B-ins)	73.6	51.0	43.2	20.4	28.5	15.0	20.0	36.0
TRIMIX (ENT) (1.5B-cpt + 3B-ins)	65.5	48.5	44.0	20.3	28.5	16.6	13.5	33.8
TRIMIX (PPL) (1.5B-cpt + 3B-ins)	81.9	59.5	46.2	15.2	37.3	19.6	30.0	41.4
TRIMIX (Upper Bound) (1.5B-cpt + 3B-ins)	85.3	59.5	47.4	22.1	37.3	20.9	34.0	43.8
Qwen2.5-7B-ins	85.1	57.0	47.8	18.2	26.4	11.7	38.0	40.6
Proxy Tuning (1.5B-cpt + 7B-ins)	89.3	58.0	47.9	14.4	32.2	13.1	15.5	38.6
TRIMIX (ENT) (1.5B-cpt + 7B-ins)	90.7	59.0	48.4	18.9	32.2	18.9	15.5	40.5
TRIMIX (PPL) (1.5B-cpt + 7B-ins)	83.7	62.5	50.9	15.2	38.2	17.6	30.5	42.6
Qwen2.5-14B-ins	89.9	61.5	57.7	22.4	30.1	13.0	25.7	42.9
Proxy Tuning (1.5B-cpt + 14B-ins)	91.5	55.5	53.1	22.1	36.3	17.2	6.5	40.3
TRIMIX (ENT) (1.5B-cpt + 14B-ins)	95.0	55.5	53.1	22.1	36.3	19.6	7.5	41.3
TRIMIX (PPL) (1.5B-cpt + 14B-ins)	92.9	66.0	55.5	17.1	38.3	21.3	33.0	46.3

Table 16: Scores (%) of different methods on the **Uyghur** tasks of MiLiC-Eval, using the Qwen2.5 models. **Topic CLS** refers to topic classification. **Read. Comp.** refers to Reading Comprehension. **Resp. Sel.** refers to Response Selection. **Title Gen.** refers to Title Generation. **MT** refers to Machine Translation. xx2en denotes translation from LRLs to English. en2xx denotes translation from English to LRLs. **Math** refers to Math Reasoning.

Model	Topic CLS	Read. Comp.	Resp. Sel.	Title Gen.	MT xx2en	MT en2xx	Math	Avg.
Qwen2.5-1.5B-base	54.2	46.5	35.6	16.9	11.0	5.8	8.7	25.5
Qwen2.5-1.5B-cpt	57.3	52.2	42.4	23.0	27.8	14.5	19.2	33.8
Qwen2.5-1.5B-ins	47.7	38.8	34.0	14.7	13.6	8.8	6.2	23.4
Contrastive Decoding	55.6	51.5	40.8	16.4	29.4	16.3	14.5	32.0
Model Merging	53.5	53.2	42.9	22.9	28.3	13.5	22.5	33.8
Qwen2.5-3B-ins	59.1	38.5	38.2	2.2	16.7	2.9	4.7	23.2
Proxy Tuning (1.5B-cpt + 3B-ins)	54.4	54.0	41.3	15.0	22.9	12.5	6.0	29.4
TRIMIX (ENT) (1.5B-cpt + 3B-ins)	58.7	50.5	37.1	15.0	22.9	13.3	7.5	29.3
TRIMIX (PPL) (1.5B-cpt + 3B-ins)	65.3	54.5	46.4	22.6	28.6	14.0	23.0	36.4
TRIMIX (Upper Bound) (1.5B-cpt + 3B-ins)	68.8	55.0	47.4	23.0	28.6	14.6	24.0	37.4
Qwen2.5-7B-ins	64.8	51.7	40.1	13.3	21.0	1.1	21.3	30.5
Proxy Tuning (1.5B-cpt + 7B-ins)	67.1	53.5	38.6	21.3	26.0	13.6	16.0	33.7
TRIMIX (ENT) (1.5B-cpt + 7B-ins)	74.4	53.5	38.6	21.3	26.0	13.6	16.0	34.8
TRIMIX (PPL) (1.5B-cpt + 7B-ins)	76.0	58.0	45.5	21.0	29.6	13.7	23.0	38.1
Qwen2.5-14B-ins	69.4	60.0	47.7	13.7	24.7	5.7	21.0	34.6
Proxy Tuning (1.5B-cpt + 14B-ins)	79.4	56.5	46.9	24.8	29.5	14.7	9.0	37.2
TRIMIX (ENT) (1.5B-cpt + 14B-ins)	77.6	56.0	46.0	20.8	29.5	15.3	5.5	35.8
TRIMIX (PPL) (1.5B-cpt + 14B-ins)	76.8	57.5	49.1	20.7	30.3	14.7	21.0	38.6

Table 17: Scores (%) of different methods on the **Kazakh** tasks of MiLiC-Eval, using the Qwen2.5 models. **Topic CLS** refers to topic classification. **Read. Comp.** refers to Reading Comprehension. **Resp. Sel.** refers to Response Selection. **Title Gen.** refers to Title Generation. **MT** refers to Machine Translation. xx2en denotes translation from LRLs to English. en2xx denotes translation from English to LRLs. **Math** refers to Math Reasoning.

Model	Topic CLS	Read. Comp.	Resp. Sel.	Title Gen.	MT xx2en	MT en2xx	Math	Avg.
Qwen2.5-1.5B-base	32.6	38.5	32.7	10.1	10.1	7.0	9.3	20.0
Qwen2.5-1.5B-cpt	39.8	42.7	28.7	15.1	10.3	9.8	4.2	21.5
Qwen2.5-1.5B-ins	36.6	32.7	31.5	3.9	12.6	7.3	7.3	18.9
Contrastive Decoding	33.1	37.0	24.8	10.8	10.1	9.6	1.0	18.1
Model Merging	29.4	46.5	30.5	14.7	9.9	9.8	5.8	20.9
Qwen2.5-3B-ins	37.6	35.3	35.5	10.2	13.4	6.3	10.0	21.2
Proxy Tuning (1.5B-cpt + 3B-ins)	33.5	42.0	35.9	8.3	14.2	9.3	5.0	21.2
TRIMIX (ENT) (1.5B-cpt + 3B-ins)	35.9	42.0	38.3	8.3	14.2	10.5	8.5	22.5
TRIMIX (PPL) (1.5B-cpt + 3B-ins)	31.9	42.0	33.4	11.2	11.5	9.9	8.0	21.1
TRIMIX (Upper Bound) (1.5B-cpt + 3B-ins)	43.5	45.5	40.1	11.3	14.2	10.7	12.0	25.3
Qwen2.5-7B-ins	37.8	44.0	40.4	9.2	14.1	4.0	11.2	22.9
Proxy Tuning (1.5B-cpt + 7B-ins)	41.9	42.5	39.8	6.7	15.2	1.2	4.0	21.6
TRIMIX (ENT) (1.5B-cpt + 7B-ins)	43.1	41.5	38.3	9.9	15.2	9.5	3.0	22.9
TRIMIX (PPL) (1.5B-cpt + 7B-ins)	38.3	46.0	38.8	12.0	12.2	9.8	8.0	23.6
Qwen2.5-14B-ins	41.7	45.0	43.7	10.4	15.6	0.7	15.3	24.6
Proxy Tuning (1.5B-cpt + 14B-ins)	51.4	42.5	45.0	10.6	16.2	7.6	1.5	25.0
TRIMIX (ENT) (1.5B-cpt + 14B-ins)	52.6	44.0	45.5	10.4	16.2	9.2	3.0	25.8
TRIMIX (PPL) (1.5B-cpt + 14B-ins)	55.8	47.5	38.8	12.7	11.6	9.9	5.5	25.9

Table 18: Scores (%) of different methods on the **Mongolian** tasks of MiLiC-Eval, using the Qwen2.5 models. **Topic CLS** refers to topic classification. **Read. Comp.** refers to Reading Comprehension. **Resp. Sel.** refers to Response Selection. **Title Gen.** refers to Title Generation. **MT** refers to Machine Translation. xx2en denotes translation from LRLs to English. en2xx denotes translation from English to LRLs. **Math** refers to Math Reasoning.