Zero-Shot Cross-Lingual Transfer using Prefix-Based Adaptation

Snegha A[‡], **Sayambhu Sen**[§], **Piyush Singh Pasi**[§], **Abhishek Singhania**[§],
Preethi Jyothi[‡]

[‡] Indian Institute of Technology Bombay, India,
[§]Amazon Alexa
{23m2160,pjyothi}@iitb.ac.in, {sensayam,piyushpz,mrabhsin}@amazon.com

Abstract

With the release of new large language models (LLMs) like Llama and Mistral, zero-shot cross-lingual transfer has become increasingly feasible due to their multilingual pretraining and strong generalization capabilities. However, adapting these decoder-only LLMs to new tasks across languages remains challenging. While parameter-efficient fine-tuning (PeFT) techniques like Low-Rank Adaptation (LoRA) are widely used, prefix-based techniques such as soft prompt tuning, prefix tuning, and Llama Adapter are less explored, especially for zero-shot transfer in decoder-only models. We present a comprehensive study of three prefix-based methods for zero-shot cross-lingual transfer from English to 35+ high- and low-resource languages. Our analysis further explores transfer across linguistic families and scripts, as well as the impact of scaling model sizes from 1B to 24B. With Llama 3.1 8B, prefix methods outperform LoRA-baselines by up to **6%** on the Belebele benchmark. Similar improvements were observed with Mistral v0.3 7B as well. Despite using only 1.23M learning parameters with prefix tuning, we achieve consistent improvements across diverse benchmarks. These findings highlight the potential of prefix-based techniques as an effective and scalable alternative to LoRA, particularly in low-resource multilingual settings.

1 Introduction

Large language models (LLMs) exhibit strong multilingual and zero-shot generalization abilities due to exposure to diverse pretraining data. Nonetheless, cross-lingual transfer remains challenging given the linguistic diversity and complexity of adapting large models efficiently without significant computational overhead.

To address the high computational and memory costs of full model finetuning, recent advances in parameter-efficient finetuning (PeFT) techniques focus on updating only a small subset of model parameters while keeping the majority of the pretrained weights frozen. This design significantly reduces the adaptation cost and makes large-scale models more practical for multilingual and domain-specific applications. Methods such as Low-Rank Adaptation (LoRA) (Hu et al., 2022) and instruction-tuned adapters have shown promising results in efficiently tailoring models to new tasks without requiring extensive resources. Among the various PeFT techniques, prefix-based approaches like soft prompting (Lester et al., 2021; Liu et al., 2024a) and prefix tuning (Li and Liang, 2021) are particularly compelling because they introduce learnable components either at the input or within the transformer stack, enabling flexible task adaptation without altering the underlying architecture of the model.

While these prefix-based techniques have been shown to be effective in monolingual scenarios and task-specific settings, their potential in facilitating zero-shot cross-lingual transfer is under-explored. This is especially relevant for decoder-only LLMs, which are increasingly being deployed in multilingual environments. Unlike encoder-decoder models that have been more thoroughly studied for transfer across languages, decoder-only models present unique challenges due to their reliance on autoregressive decoding. Understanding whether prefix-based PeFT methods can enhance zero-shot cross-lingual performance in such models has not been previously studied in detail.

In this work, we provide the first systematic study of prefix-based PeFT methods for zero-shot cross-lingual transfer in *decoder-only LLMs*. Our contributions can be summarized as follows:

- We evaluate prefix-based adaptation on models ranging from 1B parameters to large-scale 24B models to show the effectiveness of prefix tuning in multilingual transfer across models

Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025), pages 385–396
November 8-9, 2025 ©2025 Association for Computational Linguistics

of varying sizes.

- Our study spans four well-recognized multilingual benchmarks – XQUAD, XNLI, Belebele and MGSM – to compare the performance of LoRA and prefix-based tuning.

- We provide a detailed comparison of prefix-based methods (soft prompts, prefix tuning, LLaMA-Adapter) against LoRA and full fine-tuning[1], systematically analyzing their strengths and limitations across tasks and 35+ high- and low-resource languages. Additionally, we investigate transfer patterns across linguistic families and scripts.

Together, our findings position prefix-based adaptation as a lightweight yet powerful strategy for cross-lingual and reasoning-oriented applications, particularly in resource-constrained multilingual settings.

2 Related Work

Cross-lingual transfer is a key challenge in multilingual NLP. It is traditionally tackled through full fine-tuning of multilingual models. However, with large decoder-only LLMs like Llama and Mistral, full fine-tuning is costly, leading to PeFT approaches. LoRA (Hu et al., 2022) introduces low-rank trainable matrices into frozen weights to reduce training overhead. Alternatively, prefix-based methods either add learnable tokens at the input layer (Lester et al., 2021; Liu et al., 2024a) or to attention keys and values at each layer (Li and Liang, 2021), enabling efficient task adaptation.

Soft prompt tuning has been extensively studied for cross-lingual transfer in encoder and encoder-decoder models, particularly in classification tasks. For instance, (Philippy et al., 2024) demonstrated that soft prompts can generalize better across languages with fewer parameters, following the “less is more" principle. Similarly, (Philippy et al., 2025) utilized multilingual verbalizers and contrastive label smoothing to further enhance cross-lingual classification. Recent work such as (Vykopal et al., 2025) introduced language-specific soft prompts specifically designed for transfer learning, showing that combining language-specific and task-specific prompts improves generalization. However, these prior works predominantly used multilingual encoder-only and encoder-decoder models, and appended prefix tokens only to the input.

As soft prompts have several limitations in effectively adapting models to new tasks, prefix tuning emerged as a promising approach. Cross-lingual alignment through prompt-based pretraining, as proposed by (Tu et al., 2024), further improved intent classification and slot-filling performance but it is not a zero-shot setting (as in our work). A recent variant of prefix tuning is LLaMA Adapter (Dubey et al., 2024) that introduced zero-initialized attention mechanisms for efficient prefix training and achieved strong instruction-following capabilities; however, they did not evaluate on any multilingual benchmarks. A related line of work has focused on extending prefix tuning to instance-specific adaptation based on the input prompt for improved model performance (Liu et al., 2024c; Jiang et al., 2022; Liu et al., 2024b; Zhu et al., 2024).

Few comparative studies have examined parameter-efficient tuning for multilingual settings, and most have been restricted to encoder-only models or small decoder-only models with only a few million parameters. For instance, (Zhao and Schütze, 2021) systematically compared discrete prompting, soft prompting, and fine-tuning on the few-shot multilingual NLI task using XLM-RoBERTa-base. Similarly, (Tu et al., 2022) compared prompt tuning with fine-tuning across diverse NLU tasks on XLM-R and mBERT. (Tu et al., 2022) evaluate prefix tuning on the encoder-only XLM-R model and showed its effectiveness over full fine-tuning in zero-shot cross-lingual transfer. Tu et al. (2022) investigated a decoder-based multilingual model (XGLM), but their analysis was limited to a single small model. They showed that prompt tuning can sometimes surpass fine-tuning, particularly for low-resource languages, although performance remained highly sensitive to the underlying tokenization scheme. Our work significantly extends their analysis to large decoder-only LLMs and presents a comprehensive comparison of multiple prefix-based methods, including soft prompts, prefix tuning, and LLaMA Adapter.

3 Methodology

Low-Rank Adaptation (LoRA). LoRA (Hu et al., 2022) is a parametric fine-tuning technique that has become one of the most popular approaches to enable cross-lingual transfer in LLMs. It introduces trainable low-rank matrices, typically

[1] Due to computational limitations, full fine-tuning is restricted to the SQuAD dataset on Llama 3.1 8B.

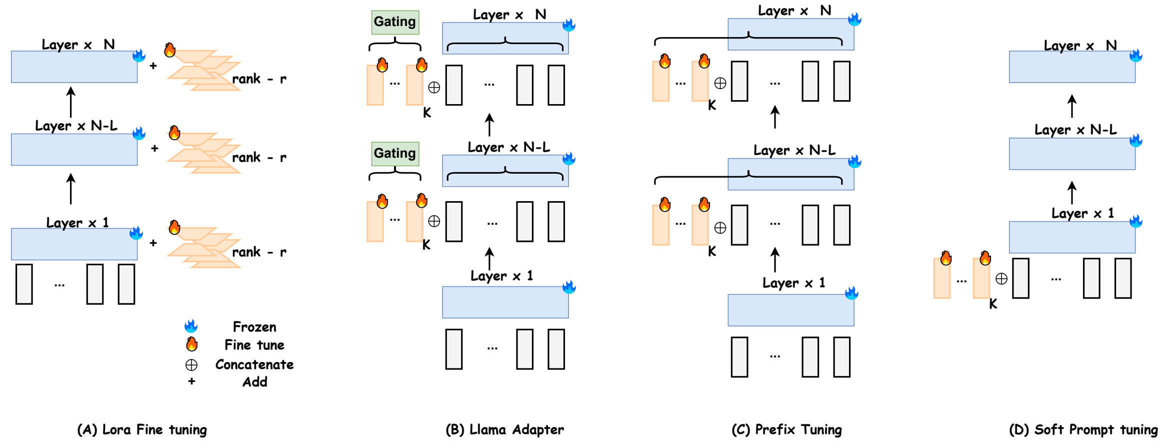


Figure 1: Schematic representation of: (A) LoRA fine-tuning and prefix-based methods, (B) Llama Adapter, (C) Prefix tuning, and (D) Soft prompt tuning.

in the query, key, and value projections, while keeping the base model frozen. These learned matrices are added to the original weights during inference. Unlike prefix-based methods, LoRA directly modifies the model parameters. A standard cross-lingual transfer setup involves fine-tuning the model using LoRA on task-specific English data and evaluating it on the target language of interest. Formally, let $W \in \mathbb{R}^{d\times k}$ be a pretrained weight matrix of a projection layer (e.g., W_q, W_k, W_v). Instead of updating W directly, LoRA parameterizes the weight update as a product of two low-rank matrices:

$$\Delta W = BA, \quad A \in \mathbb{R}^{r\times k}, B \in \mathbb{R}^{d\times r}, \tag{1}$$

where $r \ll \min(d, k)$ is the rank of the adaptation. The modified projection becomes:

$$W' = W + \Delta W = W + BA. \tag{2}$$

Given an input hidden state $h \in \mathbb{R}^k$, the output of the adapted projection layer is computed as:

$$y = W'h = Wh + BAh. \tag{3}$$

Here, only A and B are trainable, while W remains frozen. This formulation enables efficient fine-tuning by reducing the number of trainable parameters and allowing task-specific adaptation without updating the full weight matrices.

Prefix Tuning. Given an LLM, prefix tuning (Li and Liang, 2021) introduces a set of learnable prefix tokens to all layers of the transformer. In our implementation, we only append the learnable prefixes to the final L layers of the transformer. The main intuition is that these prefix tokens act as additional context vectors that the model can attend to. These vectors guide the model toward task-specific behavior, while the pretrained parameters of the LLM remain frozen.

Formally, let $P_l \in \mathbb{R}^{K\times d}$ denote the learnable prefix tokens at layer l, where K is the number of prefix tokens and d is the embedding dimension. We consider the computation for the $(M+1)$-th token, denoted by $t_l \in \mathbb{R}^{1\times d}$.The layer's input hidden states (including the current token) are represented as $H_l \in \mathbb{R}^{(M+1)\times d}$. Each attention head operates on these hidden states using projection matrices $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$.

The query vector corresponding to the current token t_l is computed using the frozen projection matrix W_q:

$$Q_l = t_l\, W_q \in \mathbb{R}^{1\times d}$$

The keys and values corresponding to the input sequence (H_l) are also computed using the frozen projection matrices W_k and W_v:

$$K_l^H = H_l\, W_k, \quad V_l^H = H_l\, W_v$$

The key idea of prefix tuning is the concatenation of the learnable prefix parameters with keys and values derived from the input.

$$P_l^K = P_l\, W_k, \quad P_l^V = P_l\, W_v$$

$P_l^K \in \mathbb{R}^{K\times d}$ and $P_l^V \in \mathbb{R}^{K\times d}$ denote the learnable prefix keys and values of layer l, respectively. The final keys and values at layer l become:

$$K_l = [P_l^K; K_l^H], \quad V_l = [P_l^V; V_l^H]$$

Method	en	hi	el	vi	sw	bg	th	ar	de	es	fr	ru	tr	zh	ur	Avg
Base Model	53.8	48.0	51.0	49.7	45.9	52.0	48.0	48.6	50.4	51.8	49.6	50.8	50.0	50.1	48.7	49.9
LoRA$_4$	90.3	70.1	74.5	77.5	60.2	73.0	72.4	71.5	77.8	79.2	80.2	73.6	73.2	77.9	65.0	74.4
Soft Prompts	84.3	67.9	54.2	72.7	51.4	51.4	66.1	63.2	52.3	57.2	59.0	59.4	58.4	41.4	62.6	60.1
Llama Adapter	93.4	74.5	**79.8**	**79.2**	59.6	78.4	76.0	**76.5**	**83.8**	83.7	84.6	79.6	**75.9**	**81.2**	**71.8**	78.5
Prefix Tuning	**93.9**	**76.5**	79.4	79.1	**60.3**	**79.4**	**76.2**	75.7	83.5	**84.4**	**85.0**	**79.9**	75.5	79.9	71.7	**78.7**

Table 1: LLaMA 3.1 8B performance (accuracy) on XNLI benchmark comparing LoRA and prefix based adaption methods.The best performance for each language is shown in **bold**, and the second-best is underlined.

Method	en	hi	el	vi	ar	de	es	ro	ru	th	tr	zh	Avg
Base Model	79.3	59.3	60.5	71.2	59.4	68.5	67.6	68.5	60.3	63.2	62.5	59.5	65.0
LoRA$_4$	86.2	66.1	72.0	75.3	68.9	76.4	78.0	78.0	72.0	**75.8**	69.1	71.7	74.1
Soft Prompts	54.5	10.8	27.9	45.5	25.8	42.2	52.0	48.2	32.2	11.2	36.8	16.1	33.6
Llama Adapter	89.4	75.1	76.9	79.8	**72.1**	82.4	83.2	82.6	**78.4**	71.6	**73.3**	72.3	**78.1**
Prefix Tuning	**90.2**	**75.7**	**78.4**	**79.3**	70.4	**82.8**	**84.2**	**83.5**	76.9	70.9	72.6	**72.5**	**78.1**

Table 2: Llama 3.1 8B performance (F1 score) on XQUAD benchmark comparing LoRA and prefix based adaption methods. The best performance for each language is shown in **bold**, and the second-best is underlined.

K_l and V_l are expanded matrices encompassing both the learned prefixes and the input sequence.

The attention scores are obtained by comparing the query Q_l against the concatenated keys K_l:

$$S_l = \frac{Q_l K_l^T}{\sqrt{d}} \in \mathbb{R}^{1\times(K+M+1)}. \quad (4)$$

The attention distribution is computed by applying the softmax function, which weights the contributions of both the prefix and the input tokens:

$$A_l = \text{softmax}(S_l) = \left[A_l^P, A_l^H\right] \in \mathbb{R}^{1\times(K+M+1)},$$

where A_l^P represents the attention weights over the learned prefixes and A_l^H represents the weights over the input sequence.

Finally, as is typically done in transformer models, the attended output representation at layer l is computed as a weighted sum of the concatenated values V_l, followed by an output projection:

$$t_l^o = (A_l V_l)\, W_o \in \mathbb{R}^{1\times d}, \quad (5)$$

where W_o is the output projection matrix. In this way, prefix tuning directly modifies the attention mechanism by injecting learned keys and values (P_l^K, P_l^V), steering the model's representations without modifying the base model weights.

Llama Adapter. The Llama Adapter (Zhang et al., 2024) builds upon the principles of prefix tuning but introduces an important modification to stabilize training in large-scale LLMs. Specifically, it replaces the standard attention mechanism with a zero-initialized variant. This mitigates instabilities that often arise from randomly initialized prefix tokens in the early stages of fine-tuning. To further enhance stability, a learnable gating mechanism is introduced, allowing the model to gradually scale the influence of prefix tokens during optimization. The gated attention scores are given by:

$$A_l^g = \left[\text{softmax}(S_l^K)\cdot\tanh(g_l),\ \text{softmax}(S_l^{M+1})\right] \quad (6)$$

where the attention scores can be split into contributions from the learnable prefix S_l^K and the original sequence S_l^{M+1}. g_l is a learnable scalar gating that adaptively controls the contribution of the prefix tokens. Finally, the output representation t_l^o is obtained using the same formulation in Equation 5. By weighting the prefix contributions using a learned gate, Llama Adapter ensures stable and effective adaptation of decoder-only LLMs.

Soft Prompts. Soft prompts (Lester et al., 2021; Liu et al., 2024a) involve prepending learnable continuous embeddings to the input, serving a similar goal as manual prompts. However, instead of manually selecting discrete prompts, soft prompting optimizes a continuous set of embeddings that serve as the prompt. This allows the model to learn how to best steer its behavior through gradient-based updates to the soft prompts.

Let $S \in \mathbb{R}^{K\times d}$ represent the learnable soft prompt embeddings, where K denotes the number of prompt tokens and d is the hidden dimension. Given an input sequence T, the modified input $\tilde{T}$ is obtained by prepending the soft prompts:

$$\tilde{T} = [S; T] \quad (7)$$

where $[;]$ denotes concatenation. The sequence $\tilde{T}$ is then passed through the transformer as usual,

Method	**en**	**th**	**zh**	**sw**	**fr**	**bn**	**de**	**te**	**ja**	**es**	**ru**	**Avg**
Base Model	50.4	23.2	27.6	**13.2**	28	**16.4**	26	**12.4**	16.8	34.8	30	25.34
LoRA$_4$	36.8	16.8	27.6	7.6	25.2	4.8	22.8	0.8	19.2	24	27.2	19.34
Llama Adapter	**53.6**	18.4	32.4	8	32.8	9.6	33.6	2	25.2	35.6	32	25.74
Prefix Tuning	52.8	**26**	**37.6**	10.8	**34**	12.8	**41.2**	6.4	**25.6**	**37.6**	**39.2**	**29.45**

Table 3: Llama 3.1 8B performance (maj@1) on MGSM benchmark comparing LoRA and prefix based adaption methods. The best performance for each language is shown in **bold**, and the second-best is underlined.

with S updated via gradient-based optimization during fine-tuning. Unlike prefix tuning, which injects key-value pairs at every transformer layer, soft prompting only modifies the input embeddings.

4 Experiments

Models. All experiments are conducted on Llama 3.1 (8B) (Dubey et al., 2024) and Mistral v0.3 (7B) (Jiang et al., 2023).To study the effect of model scaling, we additionally evaluate smaller and larger variants - Llama 3.2 (1B) and Mistral Small (24B), respectively. The Llama 3.1 and 3.2 series, developed by Meta, comprise multilingual large language models. Mistral v0.3 (7B) is an updated release from Mistral AI with an extended vocabulary compared to Mistral v0.1. Notably, Mistral Small (24B) establishes a new benchmark in the "small" LLM category (under 70B) by offering improved multilingual capabilities and a larger vocabulary. We have limited our experiment to the base model variants only.

Datasets. We evaluate on three widely-used cross-lingual benchmarks, each targeting a distinct aspect of language understanding: XQUAD (Artetxe et al., 2019) for cross-lingual question answering, XNLI (Conneau et al., 2018) for cross-lingual natural language inference, and Belebele (Bandarkar et al., 2024) for cross-lingual machine reading comprehension. We also evaluate on the MGSM (Shi et al., 2023) benchmark to assess the reasoning capabilities of large language models in multilingual settings.

Training Details We fine-tune prefix-based adaptation methods and LoRA with rank 4 using the English SQuAD training set for XQUAD containing 87.6K samples and a subset of the English NLI training data containing 100K samples for XNLI evaluations. For Belebele, we use their suggested training set containing 67.5K English samples. Finally, we use the GSM8K English training dataset with 7.47K samples (Cobbe et al., 2021) and evaluate on MGSM. All the datasets are publicly available; more training details are in Appendix A.

We experimented with learning rates (3e-3, 1e-3 and 3e-4), epochs (2,3,5), and weight decay (0.02,0.04,0.1), and report the best performance for each model. We used a learning rate of 3e-3, 2 epochs, and a weight decay of 0.02. For XNLI, we sampled 1,000 instances per language for evaluation due to computational constraints. Since XQUAD does not provide a separate test set, we evaluated on the full validation set, which includes approximately 1.19K samples per language. Finally for Belebele, we evaluated on 23 languages, where each language has 900 samples. All experiments were conducted on a single NVIDIA A100 80GB GPU.

5 Analysis and Ablations

Comparison with LoRA Fine-Tuning. Tables 1 and 2 (and Tables 14 and 15 in the Appendix) shows the performance of Llama 3.1 and Mistral v0.3 models across various tuning strategies, including LoRA, soft prompt tuning, prefix tuning, and Llama adapters on the XNLI and XQUAD datasets. To ensure fair comparisons, the number of trainable parameters in LoRA was matched with those of the prompt-based methods by setting $r = 4$ and $\alpha = 8$. The results show that prefix-based methods consistently outperform LoRA on both LLama 3.1 8B and Mistral v0.3 7B with English as the source language. This highlights the ability of prefix-based tuning for effective multilingual adaptation, even with as little as **1.23M** model parameters being trained.

We observe consistent improvements from prefix tuning across all benchmarks. Using Llama 3.1 (8B), prefix tuning achieves up to **28%** higher accuracy on XNLI, **13%** higher F1 on XQUAD, and **18%** higher accuracy on Belebele compared to the base model. Moreover, it provides additional gains of up to **4–6%** over LoRA, as shown in Tables 1, 2, 4a, and 4b. Similar trends are observed for Mistral, with consistent improvements reported in Tables,

Table 4: Overall Llama 3.1 8B performance (accuracy) on the Belebele benchmark, grouped by script and family. Best performance is in **bold**, second-best is underlined.

Script	**Language**	**Base Model**	**LoRA$_4$**	**Soft Prompt**	**Llama Adapter**	**Prefix tuning**
Cyrillic	Kyrgyz	37.2	52.9	59.3	60.5	**64.2**
	Russian	50.4	81.0	86.1	87.7	**88.1**
	Serbian	48.7	71.7	81.1	**81.9**	81.5
Burmese	Burmese	30.9	36.2	43.3	45.1	**48.4**
	Shan	31.1	28.0	30.0	29.0	**33.0**
Latin	Swati	30.2	34.3	33.4	34.3	**34.5**
	Sundanese	35.3	47.1	52.3	56.4	**57.8**
	Bambara	28.4	**34.3**	33.1	32.2	32.2
Arabic	Sindhi	36.9	46.4	51.1	53.3	**55.8**
	Egyptian Arabic	40.1	57.6	65.2	68.4	**68.7**
	Western Persian	47.5	72.9	79.6	81.4	**82.2**
Ethiopic	Amharic	30.5	34.7	37	34.9	**37.8**
	Tigrinya	24	29.2	29.7	28.1	**29.8**

(a) Grouped by language **script**.

Family	**Language**	**Base Model**	**LoRA$_4$**	**Soft Prompting**	**Llama Adapter**	**Prefix tuning**
Turkic	Kazakh	38	53.8	61.8	63.9	**64.2**
	Kyrgyz	37.2	52.9	59.3	60.5	**64.2**
	North Azerbaijani	39.9	58.4	65.4	68.3	**68.5**
Dravidian	Kannada	35.2	46.0	59.3	59.5	**61.1**
	Malayalam	35.5	49.3	56.9	60.0	**63.9**
	Tamil	36.9	52.3	60.1	60.8	**65.3**
Afro-Asiatic	Amharic	30.5	34.7	37.0	34.9	**37.8**
	Tigrinya	24	29.2	29.7	28.1	**29.8**
	Tsonga	32.7	36.3	37.3	36.1	**39**
Indo-Aryan	Sindhi	36.9	46.4	51.1	53.3	**55.9**
	Odia	33.1	38.2	54.7	55.3	**59.1**
	Sinhala	34.2	47.8	54.8	53.8	**60.8**
Balto-Slavic	Russian	50.4	81.0	86.1	87.7	**88.1**
	Serbian	48.7	71.7	81.1	**81.9**	81.5
	Slovak	46.5	73.8	80.6	83.5	**84.3**

(b) Grouped by language **family**.

16 and 17 in Appendix D.

Effectiveness of prefix-based methods across high and low-resource languages. We further evaluate the effectiveness of prefix-based methods on languages categorised as high and low resource. Since XNLI and XQUAD benchmarks primarily span high-resource languages, we rely on the Belebele benchmark to assess performance on low-resource languages. We select 23 languages for our analysis, of which 19 are considered low-resource and 4 high-resource, as per the FLoRes dataset classification. Across both the Mistral and Llama architectures, prefix-based adaptation methods yield significant performance gains while requiring only **1.23M** parameters to be tuned. Among low-resource languages, absolute improvements range from a minimum of **2%** for Shan to a maximum of **37%** for Western Persian using Llama 3.1 8B.

Prefix tuning and LLaMA adapters typically yield better cross-lingual transfer than soft prompts, likely due to more tunable parameters. However, in low-resource scenarios like those in the Belebele benchmark, soft prompting performs comparably or better as shown in Tables 4 and Tables 16, 17 in Appendix D. This is likely due to their lightweight design that helps preserve pretrained multilingual knowledge. Overall, prefix based methods appear to leverage inherent language knowledge better than LoRA.

Influence of script and language family. From Tables 4a and 4b, we observe that while both script-wise and family-wise groupings reveal performance gains with prefix-based methods, language family appears to be a reliable indicator of adaptation success. Languages within the same family tend to benefit similarly. Script-based trends show more variability, likely influenced by resource availability and linguistic diversity within a script group. The languages in our analysis span a diverse range of families such as Turkic, Dravidian, Afro-Asiatic, Balto-Slavic, and Indo-Aryan. The scripts span Cyrillic, Burmese, Arabic, Ethiopic, and Latin. Many of these languages are typologically and morphologically distant from our source language English. Prefix-based methods show strong cross-lingual performance even across distant languages, suggesting that typological similarity to English is not essential for effective adaptation. Similar trends are observed with Mistral as well, as shown in Table 16 and 17 in Appendix D.

Method	**Params**	**Acc.**
Full Fine-tuning	$\sim 8B$	37.74
LoRA$_4$	75.50M	75.99
Llama Adapter	1.23M	78.09
Prefix tuning	1.23M	**78.11**

Table 5: Comparison of full fine-tuning and parameter-efficient methods on the XQUAD dataset using LLama 3.1 8B, reported in terms of average F1 score across all languages.

Prefix-based adaptation vs. full fine-tuning. Table 5 presents a comparison of LoRA, prefix-based methods, and full fine-tuning. Detailed language-wise results are provided in Table 11 in Appendix D. We observe that while full fine-tuning leads to improvements in English, it negatively impacts the performance of target languages when applied to decoder-only models such as Llama-3.1 8B. Due to computational constraints, we were unable to extensively tune hyperparameters to achieve

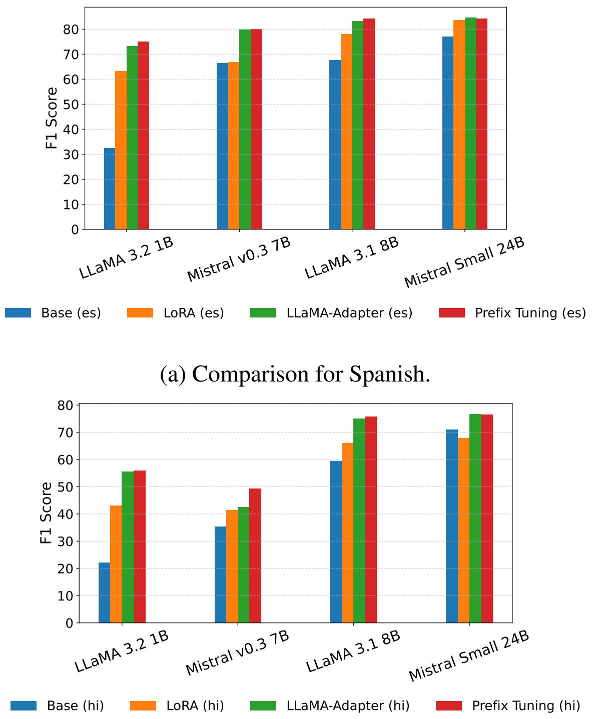
Method	LLaMA 3.2 1B	Mistral v0.3 7B	LLaMA 3.1 8B	Mistral 24B
Base Model	27.51	56.1	65.0	72.57
LoRA$_4$	56.80	59.12	74.1	70.19
Llama Adapter	64.26	65.1	**78.1**	79.70
Prefix Tuning	**64.46**	**67.2**	**78.1**	**79.94**

Table 6: Average Performance across all languages on XQUAD (F1 score) benchmark across all models comparing LoRA and prefix based adaption methods.The best performance for each language is shown in **bold**, and the second-best is underlined.

the best possible results. Overall, our findings indicate that LoRA and prefix-based methods are more effective and efficient choices for zero-shot cross-lingual transfer compared to full fine-tuning. We hypothesize that this could primarily be due to full-finetuning (on English data) leading to catastrophic forgetting in other languages.

Effect of model size on prefix-based adaptation vs. LoRA. In Figures 2b and 2a, we compare the performance of prefix-based methods against LoRA on XQUAD for Spanish and Hindi across different model sizes. We observe that both prefix tuning and LLaMA Adapter consistently outperform LoRA across all model size variations in both languages. Table 6 shows that prefix-based adaptations scale more effectively with model size, maintaining their advantage even as the underlying model grows larger. In particular, prefix tuning yields consistent improvements, thus highlighting the robustness of prefix-based approaches for multilingual transfer.

Effectiveness of prefix-based methods on MGSM. Table 3 presents results on the MGSM benchmark with Llama-3.1 8B. LLaMA Adapter and prefix tuning consistently outperform LoRA, with prefix tuning achieving the best average score (+4% over the base model). However, performance degraded for very low-resource languages like Swahili, Telugu, and Bengali. This suggests that while effective, prefix-tuning may not transfer well for complex reasoning and generation tasks without some language-specific data.



(a) Comparison for Spanish.

(b) Comparison for Hindi.

Figure 2: Comparison of prefix-based methods across model sizes against LoRA fine-tuning on XQUAD (F1 score).

Method	Params	XNLI Acc.	XQUAD F1 Score
LoRA$_4$	2.36M	74.4	74.1
LoRA$_{128}$	75.50M	76.7	76.0
Llama Adapter	1.23M	78.5	**78.1**
Prefix tuning	1.23M	**78.7**	**78.1**

Table 7: Higher Lora rank vs prefix based methods performance on XNLI and XQUAD for Llama 3.1 8B

Varying temperature/top-p during prefix-tuning. For XQUAD, we have calculated both EM (Exact match) and F1 score. From figure 3, we find that while higher temperatures and top-p values can improve F1 scores on XQUAD, they often lead to a noticeable drop in EM. This highlights a trade-off between generating more diverse predictions (captured by F1) and producing exact matches (captured by EM). The best overall trade-off is obtained at our chosen setting of temperature=0.1 and top-p=0.75.

Performance comparison of LoRA$_4$, LoRA$_{r128}$ with prefix tuning and Llama Adapters. Table 7 provides a comparative analysis of LoRA fine-tuning under two rank configurations, $r = 4$ and $r = 128$, against prefix tuning and Llama adapters. While increasing the LoRA rank from 4 to 128 substantially increases the number of trainable parameters, the resulting performance improvements are relatively modest. More importantly, our results show that parameter-efficient prefix-based approaches namely prefix tuning and Llama adapters consistently outperform LoRA, even at higher ranks. This trend is evident in both the XNLI and XQUAD benchmarks, emphasizing the

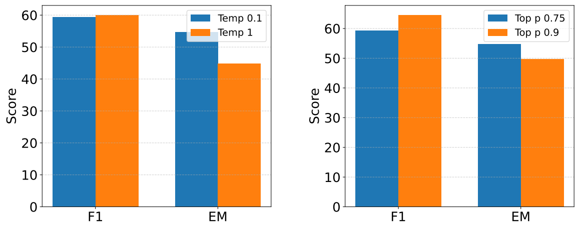


Figure 3: Varying temperature (left) and top-p (right) values using Llama 3.2 (1B) on the XQUAD task.

Layers	Params	Acc.
20	0.82M	74.5
30	1.23M	**78.7**
32	1.31M	78.0

Table 8: XNLI performance accuracy by varying number of Llama 3.1 8B layers in which prefixes are inserted.

Tokens	Params	Acc.
5	0.61M	77.8
10	1.23M	**78.7**
20	2.46M	76.0

Table 9: XNLI performance accuracy by varying number of prefix tokens in 30 Llama 3.1 8B layers.

effectiveness of prefix-based adaptation for cross-lingual transfer. These findings suggest that simply scaling LoRA with larger ranks does not necessarily close the performance gap with prefix-based methods, and the latter remains a more efficient choice for multilingual scenarios.

Impact of hyperparameter tuning on prefix-based adaptation. Prefix-based approaches are governed by two critical hyperparameters: the prefix length and the number of transformer layers in which the prefixes are inserted. In soft prompt tuning, the adaptation is constrained to the input layer, whereas in prefix tuning, prefixes can be injected across multiple layers of the model. To better understand the effect of these design choices, we systematically varied both hyperparameters. Our experiments reveal that adapting 30 out of 32 layers with a prefix length of 10 tokens provides the strongest gains across benchmarks, as summarized in Tables 8 and 9. These results highlight the sensitivity of prefix-based methods to hyperparameter configurations, and emphasize the importance of carefully selecting the number of adapted layers and prefix length to maximize performance.(For results on other models, refer to Appendix C.)

6 Conclusion

We show that prefix-based adaptation methods are a practical and efficient mechanism for cross-lingual transfer in decoder-only LLMs. Methods like soft prompting, prefix-tuning, and Llama adapters introduce learnable prefixes at different layers, while using relatively small numbers of trainable parameters. This leads to highly efficient, task-specific cross-lingual learning.

Crucially, this performance was achieved using only English training data. We hypothesize this success stems from learning language-agnostic behaviors. By adding context vectors while keeping the base model frozen, these methods preserve the LLM's inherent multilingual capabilities. In contrast, methods that alter full model weights (e.g., full fine-tuning and LoRA) suffer from catastrophic forgetting when adapted monolingually, degrading performance in unseen languages. These findings advocate for prefix-based adaptation as a robust strategy for zero-shot cross-lingual transfer.

Limitations

Our study shows that prefix-based methods yield strong zero-shot cross-lingual performance, but it has several limitations. First, due to computational constraints, our experiments were limited to 24B models; extending to larger models is a promising direction for future work. Second, our evaluations used only English as the source language. Analyzing other source languages could offer deeper insights into the methods' cross-lingual capabilities. Finally, due to computational constraints, we were unable to perform an extensive hyperparameter search for full fine-tuning. We would like to emphasize this limitation more explicitly and clarify that our intention is not to claim full fine-tuning is inherently weaker, but rather to highlight that parameter-efficient methods provide strong alternatives under realistic computational constraints. In future work, we plan to explore improving response generation for low-resource languages as seen in the MGSM benchmark and also explore more diverse response generation tasks (e.g. summarization and translation). We also plan to investigate why prefix-tuning is effective through attention visualization and representation probing.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful feedback. The last author gratefully acknowledges the generous support provided by the joint AI/ML initiative of Amazon and the Indian Institute of Technology Bombay.

References

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

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 and virtual meeting. Association for Computational Linguistics.

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

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yuezihan Jiang, Hao Yang, Junyang Lin, Hanyu Zhao, An Yang, Chang Zhou, Hongxia Yang, Zhi Yang, and Bin Cui. 2022. Instance-wise prompt tuning for pretrained language models. *arXiv preprint arXiv:2206.01958*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2024a. Gpt understands, too. *AI Open*, 5:208–215.

Yijiang Liu, Rongyu Zhang, Huanrui Yang, Kurt Keutzer, Yuan Du, Li Du, and Shanghang Zhang. 2024b. Intuition-aware mixture-of-rank-1-experts for parameter efficient finetuning. *arXiv preprint arXiv:2404.08985*.

Zequan Liu, Yi Zhao, Ming Tan, Wei Zhu, and Aaron Xuxiang Tian. 2024c. PARA: Parameter-efficient fine-tuning with prompt-aware representation adjustment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 728–737, Miami, Florida, US. Association for Computational Linguistics.

Fred Philippy, Siwen Guo, Shohreh Haddadan, Cedric Lothritz, Jacques Klein, and Tegawendé F. Bissyandé. 2024. Soft prompt tuning for cross-lingual transfer: When less is more. In *Proceedings of the 1st Workshop on Modular and Open Multilingual NLP (MOOMIN 2024)*, pages 7–15, St Julians, Malta. Association for Computational Linguistics.

Fred Philippy, Siwen Guo, Cedric Lothritz, Jacques Klein, and Tegawendé Bissyandé. 2025. Enhancing small language models for cross-lingual generalized zero-shot classification with soft prompt tuning. In *Proceedings of the 1st Workshop on Language Models for Underserved Communities (LM4UC 2025)*, pages 61–75, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Lifu Tu, Jin Qu, Semih Yavuz, Shafiq Joty, Wenhao Liu, Caiming Xiong, and Yingbo Zhou. 2024. Efficiently aligned cross-lingual transfer learning for conversational tasks using prompt-tuning. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1278–1294, St. Julian's, Malta. Association for Computational Linguistics.

Lifu Tu, Caiming Xiong, and Yingbo Zhou. 2022. Prompt-tuning can be much better than fine-tuning on cross-lingual understanding with multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5478–5485, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ivan Vykopal, Simon Ostermann, and Marian Simko. 2025. Soft language prompts for language transfer. 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 10294–10313, Albuquerque, New Mexico. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Renrui Zhang, Jiaming Han, Chris Liu, Aojun Zhou, Pan Lu, Yu Qiao, Hongsheng Li, and Peng Gao. 2024. LLaMA-adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *The Twelfth International Conference on Learning Representations*.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wei Zhu, Aaron Tian, Congrui Yin, Yuan Ni, Xiaoling Wang, and Guotong Xie. 2024. IAPT: Instance-aware prompt tuning for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14285–14304, Bangkok, Thailand. Association for Computational Linguistics.

A Prompt Templates

Training and inference prompts for all the three benchmarks we have evaluated. For MGSM, we use the 8-shot chain-of-thought prompt as in (Wei et al., 2022) (maj@1) to evaluate.

XQUAD

```
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
### Instruction:
You will answer reading comprehension questions using information from a provided passage. Extract the exact answer from the passage without modification and present it in the following structured format:
{'answer' : <Extracted Answer>}
### Input:
Context:
<context>
Question:
<question>

### Response:
{'answer':
```

Belebele

```
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
### Instruction:
The task is to perform a reading comprehension task. Given the following passage, question, and answer choices, output the number corresponding to the correct answer only.
### Input:
Passage:
<passage>
Question:
<question>
Choices:
<choices>

### Response: The correct choice number is
```

Benchmark	Languages
XNLI	en, hi, el, vi, sw, bg, th, ar, ar, de, es, fr, ru, tr, zh, ur
XQUAD	en, hi, el, vi, ar, de, es, ro, ru, th, tr, zh

Table 10: Languages used in the XNLI and XQUAD benchmarks.

XNLI

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
### Instruction:
The task is to solve Natural Language Inference (NLI) problems. NLI is the task of determining whether the inference relation between the second sentence (Hypothesis) with respect to the first sentence (Premise) is one of the following:
1. Entailment
2. Neutral
3. Contradiction
Output the relation number only.
### Input:
Premise:
`<premise>`
Hypothesis:
`<hypothesis>`

### Response: The relation number is

B Languages details

Evaluation language details included in the benchmarks are given in Tables 10, 12 and 13.

C Hyperparameter details

We insert 10 prefix tokens across 30 layers for LLaMA 3.1 8B, Mistral 7B, and Mistral 24B, while for LLaMA 3.2 1B, the tokens are inserted across all layers as it is small.For full fine-tuning, we used a batch size of 8, a learning rate of 1e-5 with a cosine learning rate scheduler, a warm-up ratio of 0.1, and trained the model for 2 epochs.Finally for LoRA fine tuning, we applied it to the Q, K, and V projection matrices across all layers.

D Complete elaborated experiment results

Language	F1	EM
ar	14.43	9.83
de	61.95	43.36
el	22.63	17.98
en	84.69	72.10
es	62.08	41.34
hi	15.23	11.76
ro	58.57	40.76
ru	18.65	10.42
th	16.13	12.35
tr	42.38	26.72
vi	43.47	25.88
zh	12.66	9.50
Avg	**37.74**	**26.83**

Table 11: Full fine tuning performance of Llama 3.1 8B on XQUAD

Language	Family
Kazakh	Turkic
Kyrgyz	Turkic
North Azerbaijani	Turkic
Kannada	Dravidian
Malayalam	Dravidian
Tamil	Dravidian
Amharic	Afro-Asiatic
Tigrinya	Afro-Asiatic
Tsonga	Afro-Asiatic
Sindhi	Indo-Aryan
Odia	Indo-Aryan
Sinhala	Indo-Aryan
Russian	Balto-Slavic
Serbian	Balto-Slavic
Slovak	Balto-Slavic

Table 12: Languages grouped by family included in Belebele

Language	Script
Kyrgyz	Cyrillic
Russian	Cyrillic
Serbian	Cyrillic
Burmese	Burmese
Shan	Burmese
Swati	Latin
Sundanese	Latin
Bambara	Latin
Sindhi	Arabic
Egyptian Arabic	Arabic
Western Persian	Arabic
Amharic	Ethiopic
Tigrinya	Ethiopic

Table 13: Languages grouped by script included in Belebele

Method	en	hi	el	vi	sw	bg	th	ar	de	es	fr	ru	tr	zh	ur	Avg
Base Model	34.3	34.7	33.8	33.6	33.4	33.8	32.8	33.6	34.1	33.8	33.5	33.6	34.2	33.9	33.6	33.8
LoRA_4	47.3	42.2	42.1	44.8	43.5	47.3	44.0	45.0	41.6	42.1	40.0	48.3	40.0	43.9	40.6	43.5
Soft Prompts	79.4	41.8	46.7	67.6	44	70.5	48.8	56.5	73.2	75.3	75.9	67.0	60.9	69.4	49.5	61.7
Llama Adapter	**92.0**	**58.1**	**64.8**	**69.3**	**46.9**	73.9	61.3	**61.7**	**79.0**	79.3	**80.6**	76.0	65.2	76.7	**55.6**	**69.4**
Prefix Tuning	90.8	56.7	61.9	**69.3**	43.4	**75.7**	**62.8**	61.5	78.8	**80.3**	79.5	**76.7**	**63.9**	**78.3**	54.5	69.0

Table 14: Mistral v0.3 7B performance (accuracy) on XNLI benchmark comparing LoRA and prefix based adaption methods.The best performance for each language is shown in **bold**, and the second-best is underlined.

Method	en	hi	el	vi	ar	de	es	ro	ru	th	tr	zh	Avg
Base Model	77.7	35.4	47.9	62.7	46.9	65.4	66.4	64.3	53.8	47.4	48.0	57.8	56.1
LoRA_4	82.5	41.37	53.52	48.0	54.1	67.1	68.2	66.7	58.8	**53.2**	51.1	64.9	59.12
Soft Prompts	72.1	1.6	19.4	42.2	18.4	61.6	62.3	59.6	49.3	10.1	48.1	18.4	38.6
Llama Adapter	**88.5**	42.5	53.4	69.1	51.1	75.9	80.0	**78.6**	**72.3**	41.3	58.3	**71.0**	65.1
Prefix Tuning	88.4	**49.3**	**60.4**	**69.5**	**55.4**	**77.4**	**80.0**	78.2	71.7	46.1	**60.9**	69.1	**67.2**

Table 15: Mistral v0.3 7B performance (F1 score) on XQUAD benchmark comparing LoRA and prefix based adaption methods.The best performance for each language is shown in **bold**, and the second-best is underlined.

Script	Language	Base Model	LoRA_4	Soft Prompt	Llama Adapter	Prefix tuning
Cyrillic	Kyrgyz	31.7	29.2	35.8	34.1	**35.5**
	Russian	57.3	62.2	83.1	**83.8**	82.3
	Serbian	55.5	60.2	79.0	**79.8**	76.5
Burmese	Burmese	28.3	23.0	**33.0**	30.8	30.7
	Shan	26.0	21.5	26.1	25.3	**27.0**
Latin	Swati	28.6	27.3	29.6	30.0	**32.0**
	Sundanese	32.1	30.5	**37.4**	35.7	35.4
	Bambara	29.3	28.3	31.3	31.2	**32.8**
Arabic	Sindhi	31.3	24.3	**31.4**	29.2	30.8
	Egyptian Arabic	39.3	35.0	**48.6**	45.1	43.7
	Western Persian	41.2	35.1	**55.4**	49.8	52.5
Ethiopic	Amharic	29.3	22.7	**31.1**	29.2	30.7
	Tigrinya	28.3	23.0	25.7	26.1	27.0

Table 16: Performance (accuracy) of Mistral v0.3 7B on the Belebele benchmark, grouped by language **script**, comparing LoRA and prefix-based adaptation methods.The best performance for each language is shown in **bold**, and the second-best is underlined.

Family	Language	Base Model	LoRA_4	Soft Prompting	Llama Adapter	Prefix tuning
Turkic	Kazakh	33.7	29.4	**38.0**	34.3	35.6
	Kyrgyz	31.7	29.2	**35.8**	34.1	35.5
	North Azerbaijani	34.7	35.2	**45.5**	42.3	42
Dravidian	Kannada	34.3	25.7	**38.1**	34.2	36
	Malayalam	31.8	25.7	**36.7**	31.8	31.4
	Tamil	34.1	29.0	39.8	36.5	**40.0**
Afro-Asiatic	Amharic	29.3	22.7	**31.1**	29.2	30.7
	Tigrinya	28.3	23.0	25.7	26.1	27.0
	Tsonga	28.4	28.5	**34.7**	33.3	33.8
Indo-Aryan	Sindhi	31.3	24.3	**31.4**	29.2	30.8
	Odia	30.5	23.6	30.3	**30.7**	30.5
	Sinhala	32.2	27.1	34.5	29.4	34.5
Balto-Slavic	Russian	57.3	62.2	83.1	**83.8**	82.3
	Serbian	55.5	60.2	79.0	**79.8**	76.5
	Slovak	52.9	58.2	**73.1**	72.8	72.3

Table 17: Performance (accuracy) of Mistral v0.3 7B on the Belebele benchmark, grouped by language **family**, comparing LoRA and prefix-based adaptation methods.The best performance for each language is shown in **bold**, and the second-best is underlined.