

# Cross-Prompt Encoder for Low-Performing Languages

Beso Mikaberidze<sup>†</sup>, Teimuraz Sagchinadze<sup>†</sup>, Simon Ostermann<sup>\*+</sup>, Philipp Müller<sup>\*◦</sup>

<sup>†</sup>Muskhelishvili Institute of Computational Mathematics, GTU (MICM),

<sup>\*</sup>Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)

<sup>+</sup> Center for European Research in Trusted AI (CERTAIN)

<sup>◦</sup> Max Planck Institute for Intelligent Systems

beso.mikaberidze@gmail.com, mueller@is.mpg.de

## Abstract

Soft prompts have emerged as a powerful alternative to adapters in parameter-efficient fine-tuning (PEFT), enabling large language models (LLMs) to adapt to downstream tasks without architectural changes or parameter updates. While prior work has focused on stabilizing training via parameter interaction in small neural prompt encoders, their broader potential for transfer across languages remains unexplored. In this paper, we demonstrate that a prompt encoder can play a central role in improving performance on *low-performing languages*—those that achieve poor accuracy even under full-model fine-tuning. We investigate a lightweight encoder paired with multi-source training on typologically diverse languages. We call this architecture-training combination the Cross-Prompt Encoder (XPE), and show that it advances the capture of abstract, transferable patterns across languages. To complement XPE, we propose a Dual Soft Prompt mechanism that combines an encoder-based prompt with a directly trained standard soft prompt. This hybrid design proves especially effective for target languages that benefit from both broadly shared structure and language-specific alignment. Text classification experiments with a transformer encoder (XLM-R) on the SIB-200 benchmark reveal a consistent trade-off: XPE is most effective for low-performing languages, while hybrid variants offer broader adaptability across multilingual settings.

## 1 Introduction

Cross-lingual task transfer (XLT) seeks to leverage supervision in one or more source languages to enable task generalization to target languages. As highlighted in a recent survey on cross-lingual alignment (Hämmerl et al., 2024), most existing approaches rely on supervising models in a single source language—typically English—before applying them to target languages. In contrast, multi-source training, where models are supervised

on multiple labeled source languages, remains relatively underexplored (Zheng et al., 2021). Yet this setup holds significant promise: By exposing the model to multiple linguistic lenses, it encourages the learning of more robust, language-agnostic representations grounded in shared structural and semantic patterns across diverse languages.

This capability becomes especially important when transferring to *low-resource* target languages—those for which even full-model fine-tuning yields suboptimal results due to a lack of data. These languages often exhibit substantial typological divergence from high-resource counterparts and lack alignment signals that typically aid transfer (Lauscher et al., 2020). In such cases, effective zero-shot transfer remains one of the most persistent challenges in multilingual NLP, as evidenced by the XTREME benchmark, which reveals consistently large performance gaps between English and many typologically diverse target languages across a range of tasks (Hu et al., 2020).

Building on prior works using *prompt encoders* (Li and Liang, 2021; Razdaibiedina et al., 2023; Liu et al., 2022b, 2024), we investigate their application in the context of multi-source XLT. We explore a setup where a small, reusable neural encoder enriches soft prompt with abstract, transferable patterns drawn from multiple, typologically diverse source languages. To resulting combination of architecture and training regime we refer as the Cross-Prompt Encoder (XPE). Unlike dynamic prompt encoder-based approaches, both the encoder and its input embeddings are static at inference time, retaining the efficiency of standard soft prompting without introducing overhead.

To complement this architecture, we also propose a Dual Soft Prompt (DUAL) mechanism that adds a directly trained standard soft prompt (SPT) alongside the encoder-based prompt. This design enables the model to incorporate both abstract, cross-lingual patterns and more language-specific

cues, offering complementary capabilities that can benefit each target language to varying degrees.

Our experiments on the SIB-200 benchmark—covering over 200 languages—demonstrate that XPE achieves strong performance on low-performing and typologically diverse target languages, while DUAL variants excel in other cases settings. Together, these findings highlight the strength of combining multilingual supervision with prompt modularity, enabling efficient XLT across a wide spectrum—from well-aligned to more challenging scenarios.

While we expect the proposed setup to generalize across architectures and tasks, this paper focuses specifically on zero-shot and fully supervised text classification with a transformer encoder backbone (XLM-R), reflecting the practical scope of this study.

Our contributions are three-fold:

1. We empirically study a parameter-efficient method that combines a soft prompt encoder with multi-source training on typologically diverse languages. This setup—referred to as the Cross-Prompt Encoder (XPE)—enhances cross-lingual task transfer (XLT) by encouraging the model to learn broadly applicable patterns. To our knowledge, this is the first study to apply prompt encoders in a hybrid, multi-source transfer setting.
2. We achieve state-of-the-art performance in both zero-shot transfer and full-data scenarios on SIB-200 text-classification task (Adelani et al., 2024). Our method outperforms zero-shot prompting models (e.g., GPT-4), prompt-based ZS-XLT methods (e.g., RoSPrompts), and full-model fine-tuning baselines (e.g., SIB-200) across a wide range of languages. It is especially effective on low-performing languages — those that remain challenging even under direct full-model fine-tuning.
3. We conduct ablation experiments to analyze the strengths of encoder-based and standard soft prompts. Our findings show that XPE is more effective in challenging low-performing scenarios, while standard soft prompts perform better when the source and target languages are closely aligned. Based on this, we introduce a Dual Soft Prompt (DUAL) mechanism that combines both, consistently yielding the best performance across multilingual

settings.

Our code and pretrained prompts are publicly available through GitHub<sup>1</sup>.

## 2 Related Work

With the rise of LLMs, a new paradigm of PEFT has emerged due to the size of models being fine-tuned (Han et al., 2024; Wang et al., 2025). The general goal in mind is to minimize the number of parameters to be trained while enhancing model performance above in-context learning and ideally approaching the performance of full-fine tuning (Liu et al., 2022a). After validating its performance in single task / language scenarios PEFTs are often modified to work within multi-language problems (Pfeiffer et al., 2020; Fu et al., 2022b).

### 2.1 Parameter-efficient Cross-lingual Adaptation

MAD-X (Pfeiffer et al., 2020), based on adapters (Houlsby et al., 2019), is one of the first methods to be successfully extended to multilingual environments. Recently, LoRA (Hu et al., 2021) was extended to cross-lingual scenarios using a method called FLARE (Borchert et al., 2025). One drawback of this method however is that all data points must be paired with their translation in the source language. LT-SFT (Anselli et al., 2022) and its more recent variation DeFT-X (Simon and Jyothi, 2025) use *Lottery Ticket Hypothesis* to employ masks in one case and in another SVD to obtain subnetworks that correspond to task and language separately and combine them to obtain cross-lingual transfer.

Major PEFT branches are viable for cross-lingual transfer, yet their zero-shot capabilities are constrained. A key limitation for approaches like MAD-X, LT-SFT, and DeFT-X is their dependence on language-specific components extracted through masked language modelling. These methods are inaccessible for languages with insufficient or non-existent unlabelled corpora, significantly limiting their utility in resource-scarce settings.

### 2.2 Soft Prompt Tuning for Cross-lingual Tasks

A recently emerging approach in parameter-efficient adaptation is to find prompts or prefixes using backpropagation, dubbed soft-prompts (Li and Liang, 2021; Lester et al., 2021). Their success

<sup>1</sup><https://github.com/bmikaberidze/XPE>

in single-task environments inspired researchers to extend soft prompts to multitask and multilingual environments. (Fu et al., 2022b).

Cross-lingual transfer can be achieved through various mechanisms, including the use of a basic soft prompt (Philippy et al., 2024), a Mixture-of-Experts approach in the case of SMoP (Choi et al., 2023), or the introduction of an explicit soft prompt translation mechanism in the case of MPT (Qiu et al., 2024). On the one hand, some researchers argue that the limited number of parameters in soft prompts enhances performance (Philippy et al., 2024). However, in many other cases, some layers increase the parameter count while keeping the width of the injected prompt relatively small (Qiu et al., 2024; Choi et al., 2023).

Soft prompt based approach can be used in zero-shot scenarios; strategies are varied too, including finding a universal prompt across multiple tasks and multiple languages (Fu et al., 2022b) in the case of Polyprompt, tweaking loss and learning procedures, or even a template/context split fusion mechanism for UniPrompt (Huang et al., 2022) and RosPrompt (Philippy et al., 2025). However, the results are difficult to compare, as they all utilize different datasets and do not necessarily employ the same method to select source and target languages.

Out of all the data sets proposed in these articles, SIB-200 contains the largest number of languages and has additional labels, including Joshi’s classification (Adelani et al., 2024). This dataset lets us explore languages usually missing from a model’s pretraining or those that generally underperform. Existing methods have limitations: UniPrompt cannot directly evaluate languages the model hasn’t encountered, and Polyprompt, though interesting, was trained on mT5 (Xue et al., 2021), making direct comparison challenging. What sets our work apart is its direct focus on underperforming languages—a gap, to our knowledge, not addressed by previous research. This distinct focus may explain why RoSPrompt underperforms in comparison to our proposed method.

### 3 Methodology

To address the challenge of zero-shot cross-lingual transfer (ZS-XLT)—particularly for low-performing languages—we investigate a reusable and parameter-efficient prompt encoder trained under a multi-source supervision regime. This setup, inspired by prior work such as P-Tuning (Liu

et al., 2022b) and Multitask Prompt Tuning (MPT) (Wang et al., 2023), is referred to as the Cross-Prompt Encoder (XPE) (see Figure 1(b)). XPE consists of a single, reusable neural module that encodes a soft prompt using supervision from multiple typologically diverse source languages. The encoder and its inputs are shared across all languages, and the encoding process induces interactions among those input embeddings. Hence, the encoded soft prompt is able to learn abstract, language-agnostic patterns, thereby enhancing transferability, especially for low-performing and poorly aligned languages. At inference time, the encoded prompt is cached and used directly, preserving the efficiency of standard soft prompt tuning.

To complement this design, we introduce a Dual Soft Prompt (DUAL) mechanism that integrates XPE with an additional, directly trained standard soft prompt (SPT) (see the Figure 1(c)). As the standard soft prompt does not involve a prompt encoder, it is expected to capture more language-specific features, which may assist in transferring to languages seen during backbone model pretraining or those closely aligned with them. The resulting DUAL setup supports robust multilingual transfer across a broad spectrum of languages—ranging from well-aligned to low-performing ones—each may benefit to varying degrees from both components.

#### 3.1 Cross-Prompt Encoder (XPE)

XPE employs a lightweight neural network that maps a small set of learnable input embeddings to outputs with the same hidden dimension as the frozen backbone model. We refer to these inputs as the *pseudo prompt*, and to the network’s output as the *encoded soft prompt*.

Importantly, the prompt encoder and pseudo prompt are used only during training. Once training is complete, the encoder transforms the pseudo prompt into a static encoded soft prompt, which is cached and prepended at inference time—avoiding any additional computation or architectural change.

Formally, the encoder is defined as  $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ , where  $\theta$  denotes the parameters of the encoder module and  $d$  is the hidden size of the backbone Transformer, corresponding to its input embedding dimension. While pseudo prompt represents a matrix  $n \times d$  where  $n$  is the number of embeddings in the pseudo prompt, the encoder can handle only one embedding at a time. The resulting vectors are concatenated to form the final encoded soft prompt.

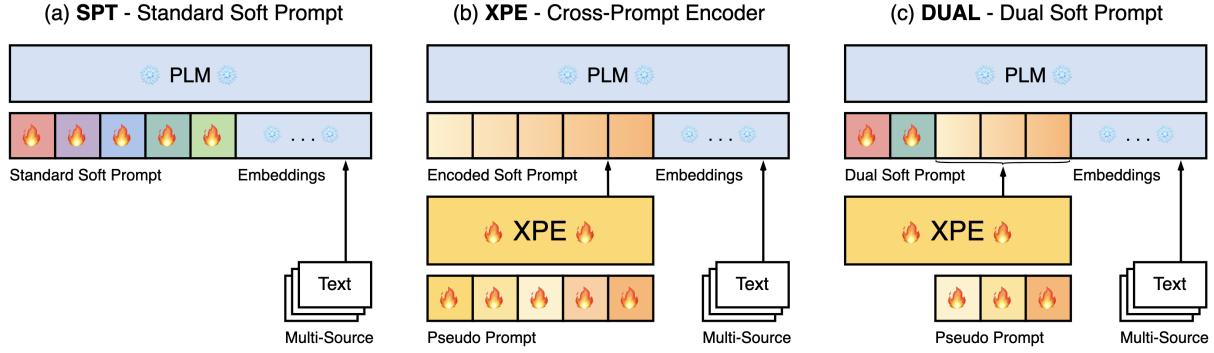


Figure 1: Architectural setup of the three methodologies during training: (a) **SPT** – Standard Soft Prompt, (b) **XPE** – Cross-Prompt Encoder, and (c) **DUAL** – Dual Soft Prompt, a hybrid combining both prior approaches. Fire and snowflake icons indicate trainable and frozen parameters, respectively.

The overall mapping from the full pseudo prompt to the encoded soft prompt can thus be expressed as  $F_\theta : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$

The pseudo prompt  $n \times d$  and encoder parameters  $\theta$  are shared across all source languages.

### 3.2 Dual Soft Prompt (DUAL)

The DUAL setup integrates XPE and standard SPT approaches, enabling the prompt to combine both encoder-based shared structure and directly learned embeddings. Specifically, we allocate a fixed number of soft prompt embeddings to two components: the first part is dedicated to a standard soft prompt, and the second to an encoded soft prompt. These two segments are concatenated—standard first, followed by encoded—as illustrated in Figure 1 and jointly tuned during training. The full soft prompt is injected at the embedding layer, while the backbone model remains frozen throughout.

Like the encoder parameters and pseudo prompt used in XPE, the standard soft prompt is shared across all source languages. So like XPE, the DUAL setup also produces a static, multilingual soft prompt, which is solely prepended to the input embeddings of the backbone model at inference time. This composite prompt preserves the overall token budget while blending both components. We experiment with two configurations: DUAL<sup>XPE-70</sup> and DUAL<sup>XPE-30</sup>. In both variants, the numbers 70 and 30 denote the percentage of soft prompt tokens allocated to the XPE component, with the remaining tokens used for the standard SPT.

## 4 Experiments

We conduct experiments on the SIB-200 multilingual text classification benchmark, focusing on both zero-shot and fully-supervised XLT scenar-

ios. The experiments are designed to assess the effectiveness of soft prompt tuning methods under diverse multi-source training setups, and to analyze performance across several meaningful target language groups, including the most challenging low-performing languages. All evaluations are conducted per target language, with aggregate results reported at the group level. All models are built upon the XLM-R large encoder and are compared against strong baselines, including full-model fine-tuning and zero-shot prompting. We also perform a detailed ablation study to isolate the contributions of each component.

### 4.1 Experimental Setup

Our experiments are based on the XLM-R large model, a transformer encoder pretrained on 100 languages. During training, the backbone remains frozen, and we optimize only a small set of parameters, that include soft prompt related parameters and the transformer classification head. The total number of trainable parameters remains under 0.3% of the full model, enabling highly parameter-efficient transfer learning (PETL).

We adopt a lightweight classification head instead of a verbalizer to ensure consistent evaluation across all ~200 languages, including those not supported by XLM-R’s tokenizer. While this departs from a “pure” soft-prompt setup, it avoids the well-known cross-lingual bias introduced by shared verbalizers (Li et al., 2023), as well as the limitations of language-specific verbalizers that only apply to tokenizer-covered languages (Philippy et al., 2025). In scenarios prioritizing plug-and-play modularity or few-shot generalization, however, a verbalizer remains a viable alternative.

We evaluate on the SIB-200 benchmark, a mul-

tilingual topic classification dataset covering 200 typologically diverse languages.

The general setup of our experiments is a multi-source cross-lingual task transfer (XLT), conducted under two levels of supervision: zero-shot and full. In the zero-shot setting, the model is trained on labeled data from the source languages and directly applied to each target language without any target supervision. In contrast, the fully-supervised setting follows a sequential XLT setup, where the model is first tuned on the multi-source data, then further tuned on labeled data from a single target language before evaluation.

## 4.2 Sources and Target Language Grouping

To study cross-lingual transfer dynamics under diverse conditions, we define several configurations for both source and target language sets. We use the following source configurations: 1. *EnArZho*: A compact, high-resource, typologically diverse set comprising English, Arabic, and Mandarin Chinese. 2. *Joshi5*: A group of seven most high-resource languages classified as 5 by Joshi et al. (Joshi et al., 2020). 3. *Seen*: The 92 languages that were included in the XLM-R pretraining corpus, representing the model’s seen-language space. Notably, each small group is a subset of bigger groups.

To better interpret transfer effectiveness, we aggregate results across four target language groups based on their relationship to XLM-R pretraining and downstream performance: 1. *All /wo Joshi5*: All SIB-200 languages excluding the Joshi5 set. 2. *Seen /wo Joshi5*: Consisting of only languages seen during XLM-R pretraining, excluding the Joshi5 set. 3. *Unseen*: Languages not included in XLM-R’s pretraining corpus. 4. *Low-Performing* - We define low-performing languages as those for which XLM-R exhibits poor downstream performance, likely due to limited or ineffective representation during pretraining. Specifically, we identify such languages in SIB-200 by referring to full fine-tuning results on XLM-R large, reported in the original benchmark and selecting those with accuracy below 60%.

We note that the *Seen* and *Unseen* groups form a disjoint partition of the full language set (*Seen + Unseen = All*), and the *Low-Performing* group is a strict subset of *Unseen*. While this overlap is not enforced by definition, it aligns with expectations that languages unseen during pretraining tend to suffer from lower downstream performance.

For our fully-supervised experiments, we eval-

uate on a representative subset of 46 target languages—23 from the *seen* group and 23 from the *unseen* group—due to the prohibitive cost of training 200 dedicated models. Languages were selected to ensure diversity across language families, scripts, and resource levels. Although the selection process did not explicitly consider performance tiers, 11 out of the 23 unseen languages in the subset are later identified as low-performing, indicating a fair and challenging distribution.

## 4.3 Methods Compared

While our focus is on soft prompt-based transfer methods, there are relatively few established baselines for this setting on large-scale multilingual benchmarks like SIB-200.

Our main method is the Cross-Prompt Encoder (XPE), a parameter-efficient soft prompt encoding approach for multilingual transfer. To isolate the contribution of the prompt encoder, we ablate it by removing the encoder from XPE, resulting in *Standard Soft Prompt Tuning (SPT)*, which corresponds to the canonical soft prompt tuning approach widely used in prior work. This variant simultaneously serves as a baseline and a direct ablation of our method.

We additionally evaluate a hybrid setup, *Dual Soft Prompting*, which combines the SPT and XPE components within a fixed prompt budget. This setup preserves the overall number of soft prompt embeddings while blending both prompt types. We experiment with two configurations: DUAL<sup>XPE-70</sup> and DUAL<sup>XPE-30</sup>, where 70 and 30 refer to the percentage of prompt embeddings allocated to XPE.

To contextualize the performance of our approach, we compare it against several baselines, including zero-shot prompting with large language models, prompt-based transfer method using a single source language, and full-model fine-tuning on the SIB-200 benchmark. All models—except the zero-shot prompting LLM baselines—are based on the XLM-R large architecture, just like ours.

For zero-shot cross-lingual transfer (ZS-XLT), we include results from several prompting-based baselines: *Phi-3.2-mini*, *GPT-3.5*, and *GPT-4*, each evaluated in a pure zero-shot setting without any task-specific tuning. We also compare against *RoSPrompt*, a recent method that combines soft and hard prompts using English as the sole source language. Although evaluated on SIB-200, it is trained on DBpedia14 (a different topic classification dataset), and relies on language-specific verbal-

Target	Source	#Source	SPT	DUAL <sup>XPE-30</sup>	DUAL <sup>XPE-70</sup>	XPE
LowPerf.	EnArZho	3	35.2 $\pm$ 2.4	<b>36.5</b> $\pm$ 2.3	36.2 $\pm$ 2.7	35.3 $\pm$ 3.2
	Joshi5	7	36.0 $\pm$ 2.3	36.8 $\pm$ 2.5	37.3 $\pm$ 2.8	<b>39.1</b> $\pm$ 2.0
	Seen	92	39.1 $\pm$ 1.7	39.3 $\pm$ 2.3	40.3 $\pm$ 2.2	<b>41.9</b> $\pm$ 1.8
Unseen	EnArZho	3	54.8 $\pm$ 2.3	<b>56.0</b> $\pm$ 2.3	55.6 $\pm$ 2.6	53.7 $\pm$ 2.8
	Joshi5	7	56.0 $\pm$ 2.0	56.4 $\pm$ 2.1	<b>57.6</b> $\pm$ 2.2	57.2 $\pm$ 1.8
	Seen	92	58.9 $\pm$ 1.4	59.5 $\pm$ 1.7	60.1 $\pm$ 1.6	<b>60.8</b> $\pm$ 1.4
Seen /wo J5	EnArZho	3	84.7 $\pm$ 1.6	<b>84.8</b> $\pm$ 1.8	84.6 $\pm$ 2.0	82.6 $\pm$ 1.8
	Joshi5	7	85.6 $\pm$ 1.3	85.3 $\pm$ 1.4	<b>86.6</b> $\pm$ 1.1	84.5 $\pm$ 1.3
All /wo J5	EnArZho	3	67.7 $\pm$ 2.0	<b>68.4</b> $\pm$ 2.1	68.0 $\pm$ 2.3	66.2 $\pm$ 2.3
	Joshi5	7	68.7 $\pm$ 1.7	68.8 $\pm$ 1.8	<b>70.0</b> $\pm$ 1.7	69.0 $\pm$ 1.6

Table 1: ZS-XLT performance (accuracy) across different target groups. Each method was trained on different source language groups. Darker yellow indicates better performance (per target group). All values include 95% two-sided confidence intervals (mean  $\pm$  half-width). J5 refers to Joshi5 languages.

izers, making the setting not directly comparable. Finally, we include the *SIB-200 ZS-XLT* baseline, corresponding to full-model fine-tuning on a single source language (English, Arabic and Chinese), followed by zero-shot evaluation on target languages.

We compare our fully-supervised multi-source XLT approach—based on parameter-efficient tuning—with the monolingual full-model fine-tuning baseline reported in the original SIB-200 benchmark. Both setups involve training a separate model for each target language; however, our method first performs multi-source training before adapting to each target language, enabling knowledge transfer across languages while updating less than 0.3% (1.6M) of the model parameters. In contrast, the SIB-200 baseline trains all model parameters on target-language supervision, without incorporating any cross-lingual signals.

Notably, we report average results over 10 and 6 random seed runs for the zero-shot and fully supervised scenarios, respectively.

#### 4.4 Implementation Details

The soft prompt length is fixed at 20 virtual embeddings. Optimization is performed using Adafactor with a fixed learning rate and a cosine schedule with restarts (2 cycles). For XPE, we use a learning rate of  $5e-5$  and weight decay of 0.1 for both the prompt encoder and the classification head. In SPT, only the soft prompt is trained with a higher learning rate of  $5e-3$  and no weight decay, while the classification head remains under the same settings as in XPE. The DUAL configuration reflects the same settings applied to its respective components. Training is conducted with a batch size of

32. Early stopping is applied after the first learning rate cycle, with a patience of 20 epochs for source training and 30 for target; if early stopping does not trigger, training stops at a hard limit of 24,000 optimization steps for source and 6,000 for target. All experiments are run on a single NVIDIA A100 GPU, with each training run taking approximately 30 minutes. We use the HuggingFace ecosystem (Wolf et al., 2020) to access the required artefacts, in accordance with the allowed scientific use.

## 5 Results

We extensively evaluate XPE and our proposed DUAL approaches in zero-shot experiments on the SIB-200 dataset (Adelani et al., 2024). We furthermore conduct an evaluation in the full fine-tuning scenario on the same dataset.

### 5.1 Zero-shot Experiments

**What mix of SPT and XPE works best?** In Table 1 we present results of different soft prompt methods and different combinations of target and source languages in a zero-shot scenario. We compare SPT with XPE as well as the two DUAL variants DUAL<sup>XPE-70</sup> and DUAL<sup>XPE-30</sup>. For the challenging set of low-performing languages, XPE achieves the best performance with 41.9 accuracy when training on all 92 seen languages (Welch’s unpaired t-test vs. SPT,  $p = 1.3 \times 10^{-5}$ ). Decreasing the proportion of XPE embeddings in favour of SPT decreases performance in this scenario. When training on only 7 source languages (Joshi5), the pattern of results is still the same, but when reducing the training languages to 3 (English, Arabic, Mandarin Chinese), utilising a mixture of SPT and

	ZS Prompting				ZS-XLT				
	Phi-3.5	GPT-3.5	GPT-4	Eng	SIB-200	RoS	SPT	DUAL <sup>XPE-70</sup>	XPE
	–	–	–		Ara	Zho	Eng	Joshi5	Seen
LowPerf.	–	22.9	22.9	33.5	33.3	33.3	–	36.0	37.3
Unseen	–	35.7	39.2	54.0	54.7	54.3	–	56.0	57.6
Seen /wo J5	49.02	55.7	68.1	86.2	86.5	86.5	67.3	85.6	86.6
All /wo J5	–	44.3	51.7	67.8	68.3	68.1	–	68.7	70.0

Table 2: Average accuracy across target language groups. The first header row indicates the general setup category, while the next two rows specify the individual methods and their corresponding source language(s). Baselines (Phi-3.5-mini, GPT-3.5, GPT-4, RoSPrompt, and SIB-200) are sourced from prior work, whereas SPT, XPE, and DUAL variants are our trained models. “J5” refers to the Joshi5 language group.

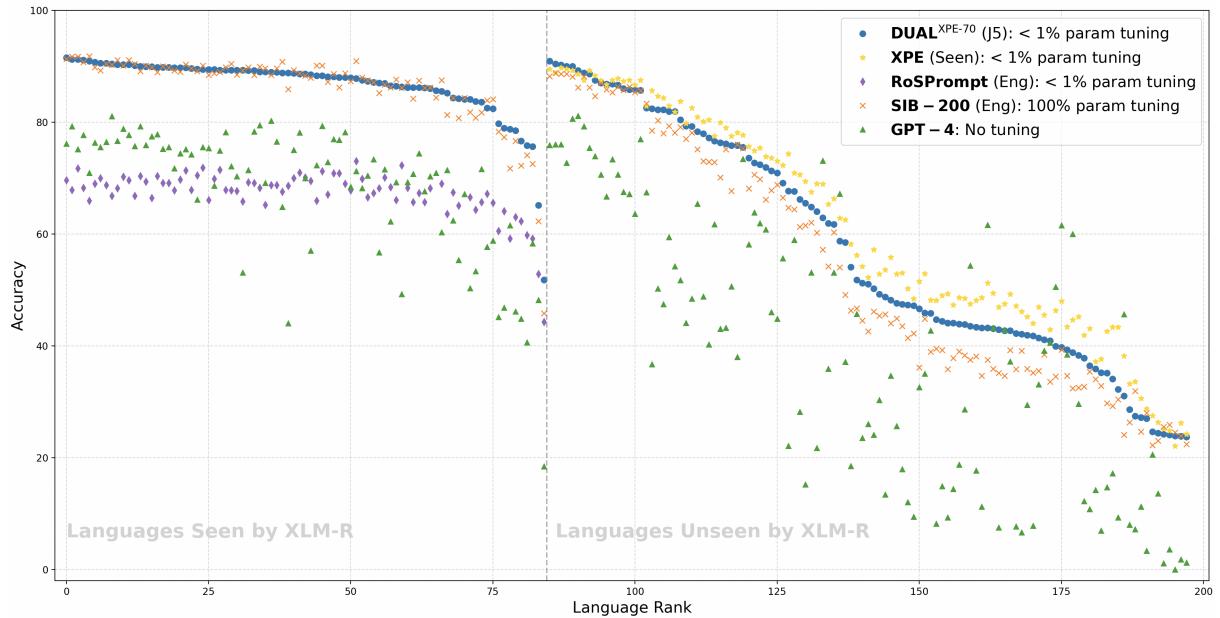


Figure 2: Comparison of different methods on the SIB-200 dataset. We group languages by whether they are seen in the pre-training corpus of XLM-R. Source language groups are provided in parentheses alongside the methods, where J5 refers to the Joshi5 group. Languages are ordered by DUAL<sup>XPE-70</sup> (J5) performance in each group. All the methods are ZS-XLT, except for GPT-4, which is ZS prompting. It should be noted that RoSPrompts used English DBpedia14 as a source dataset (a topic detection task with a different label space) and also employed language-specific verbalizers, introducing further evaluation mismatches.

XPE is more advantageous. A similar general pattern can be observed when all unseen languages are used as target languages. When training on all 92 seen languages, pure XPE reaches the best performance (60.8 accuracy). With a reduced number of training languages, mixing XPE with SPT becomes advantageous.

When considering the less challenging transfer scenarios of seen languages as targets (excluding Joshi5), the advantage of combining SPT with XPE becomes evident. In this setup, DUAL<sup>XPE-70</sup> consistently attains the highest performance. It also reaches the top result with 70.0 accuracy when evaluated across all languages (except Joshi5; Welch’s unpaired t-test vs. SPT,  $p = 0.038$ ). Per-language

results are provided in Appendix A.

**SOTA comparison.** Most importantly, our proposed approach DUAL<sup>XPE-70</sup> trained on Joshi5 outperforms all baselines in all target language configurations. Considering all languages except Joshi5 as target languages, DUAL<sup>XPE-70</sup> reaches 70.0 accuracy, followed by SPT (trained on Joshi5, 68.7 accuracy), and SIB-200 (trained on Arabic, 68.3 accuracy). For unseen languages, including the subset of low-performing languages, we reach an even higher performance using pure XPE trained on all seen languages. Here, XPE achieves an accuracy of 60.8, followed by DUAL<sup>XPE-70</sup> with 57.6 accuracy. The best result from previous work is SIB-200 trained on Arabic with 54.7 accuracy. These results

Target	#Target	SIB-200	DUAL <sup>XPE-70</sup>
Unseen	23	64.0	<b>65.1</b>
Seen	23	<b>88.3</b>	87.6
All	46	76.1	<b>76.3</b>

Table 3: Comparison of fully supervised methods. The direct full-model fine-tuning baseline is sourced from the SIB-200 paper. Our sequential XLT approach uses Joshi5 as the source language group and DUAL<sup>XPE-70</sup> as the method. Results are reported across target language groups.

underline that our approach is able to effectively integrate training signals from several source languages in challenging cross-lingual task transfer scenarios. In Figure 2 we present a plot of per-language accuracies for different methods. The improvements made by our proposed approaches are highly consistent. Only for a small number of languages our approaches are outperformed by the generally inferior GPT-4.

## 5.2 Fully-supervised Experiments

In addition to the zero-shot setting, we also evaluated our proposed approaches in a scenario where supervised data in the target language is available. For computational feasibility, we evaluate on a representative subset of 23 seen and 23 unseen languages, as described in Section 4.2. The results are shown in Table 3. Overall, our DUAL<sup>XPE-70</sup> approaches reaches a slight improvement over SIB-200. When comparing the gains for unseen versus seen target languages, we see that DUAL<sup>XPE-70</sup> particularly excels for unseen languages, whereas it is at a slight disadvantage for seen languages.

## 5.3 Additional Probing Experiment

To test our hypothesis that XPE encodes more generalizable, language-agnostic information compared to SPT, we conduct a probing experiment. We convert SIB-200 into a unified language identification dataset by discarding topic labels and using language IDs as supervision. The resulting dataset is released as LID-200 on Hugging Face.<sup>2</sup>

Specifically, we train a lightweight classifier to predict language ID from hidden representations produced by frozen soft prompts and the frozen XLM-R large backbone. We probe the soft prompts trained in zero-shot experiments. For each combination of source group (Seen, Joshi5) and method

Source	Target	XPE	SPT
Seen	LowPerf.	<b>0.78</b>	0.82
	Unseen	<b>0.76</b>	0.78
Joshi5	LowPerf.	<b>0.81</b>	0.83
	Unseen	<b>0.78</b>	0.80
	Seen /wo J5	0.91	<b>0.90</b>
	All /wo J5	<b>0.84</b>	0.85

Table 4: Probing results measuring language identifiability. Lower values indicate more language-agnostic representations. Each score is computed across 10 soft prompts and 5 random seeds, yielding 50 values per method and target-language group. Bold highlights lower identifiability.

(XPE, SPT), we use 10 soft prompts, each evaluated with 5 random seeds. This results in 50 classification scores per method and target language group, which are directly averaged to obtain a single mean score. Lower classification accuracy corresponds to lower language identifiability, indicating that the model’s representations abstract away from language-specific features. Results are shown in Table 4. We observe slight, but consistently lower identifiability for XPE compared to SPT, in five out of six cases. Importantly, the difference is more noticeable in *LowPerf.* and *Unseen* target groups, highlighting that the generalization advantage of XPE emerges specifically in more challenging transfer scenarios. For example, on low-performing languages with Seen sources, XPE achieves 0.78 accuracy versus 0.82 for SPT. For the *Seen /wo J5* target group and its superset *All /wo J5*, the small or reversed gap is unsurprising, as XLM-R already provides strong language-specific representations for seen languages, leaving little room for either method to exert additional influence.

## 6 Discussion

### 6.1 Generalization vs. Specialization

Our results reveal a consistent performance divide between XPE and SPT across different language types. XPE outperforms SPT on low-performing, typologically diverse target languages, suggesting that it is better suited for generalization. This likely stems from its encoder-based structure, which encourages abstraction and captures task-relevant patterns that generalize across languages. In contrast, SPT achieves higher scores on seen languages, where alignment with the backbone pretraining data is stronger. This indicates a tendency toward

<sup>2</sup><https://huggingface.co/datasets/mikaberidze/lid200>

specialization, where SPT embeddings memorize language-specific patterns that can be directly exploited when sufficient overlap exists between training and target data.

While SPT parameterizes soft prompt embeddings independently and optimizes them directly through the task loss, XPE computes prompt representations via a shared encoder that also mediates their optimization. Intuitively, this design promotes generalization in two complementary ways. Because the encoder applies the same parameters to each prompt embedding, it (1) induces consistency across their representations in the forward pass and discourages position-specific patterns, and (2) mediates gradient flow through shared weights in the backward pass, coupling the updates across embeddings and smoothing gradient signals, thereby encouraging more abstract and transferable representations.

Our results show that it is beneficial to combine SPT and XPE into a DUAL configuration that is able to perform well across a wide range of source and target language scenarios.

## 6.2 Language Diversity Shifts the Balance Toward Generalization

Our experiments across multiple source language configurations demonstrate that source language diversity plays a critical role in shaping cross-lingual performance. As the number of source languages increases—from 3 to 7 to 92—the benefits of generalization become more pronounced. Specifically, we observe that models with stronger generalization capacity (e.g., XPE-70 and full XPE) improve consistently with increasing diversity, often surpassing more specialized approaches like SPT. This pattern holds across both seen and unseen target groups, suggesting that language diversity amplifies the value of language-agnostic task representations. Importantly, these gains occur without exceptions across all source configurations and target groupings. This consistency highlights the universal benefit of source language diversity and supports the claim that generalization becomes increasingly crucial in multilingual transfer.

## 6.3 Alternative Explanation: Capacity Matching

Although the optimization schedule remains fixed, the number of unique training samples varies across source configurations. One might therefore attribute our findings to a capacity-matching effect:

smaller models (e.g., SPT) perform better with less data, while larger ones (e.g., XPE) benefit from greater diversity. However, the evidence instead points to architectural bias. SPT consistently performs best on seen, well-aligned languages, regardless of source diversity and size. In contrast, XPE outperforms SPT on low-performing targets across all configurations. This persistent divide suggests that inductive biases—specialization versus generalization—play a more decisive role than model size or training volume. Additionally, the consistent advantage of the DUAL setup in diverse settings suggests that combining architectural biases is more critical than model capacity alone.

## 7 Conclusion

We apply a parameter-efficient prompt encoder in a multi-source cross-lingual transfer setting. This setup, referred to as the Cross-Prompt Encoder (XPE), updates less than 0.3% of model parameters while achieving substantial gains in the most challenging scenario—zero-shot transfer to low-performing languages. To further boost adaptability, we propose a Dual Soft Prompt mechanism that combines XPE with standard soft prompts, leveraging both abstract, transferable patterns and language-specific memorization. This hybrid design enables robust multilingual transfer across a wide spectrum of target languages, each benefiting to varying degrees from the complementary strengths of both components.

## Limitations

This work focuses on a single backbone model (XLM-R), which limits conclusions about the general applicability of XPE and DUAL to other architectures, such as encoder-decoder or decoder-only models. We evaluated our method on the SIB-200 dataset. While this dataset has a large variety of languages, it is centred on a single task: multilingual topic classification. Further investigation is needed to assess generalization of our approach across different task types, including reasoning and language generation tasks. Finally, while we explore multilingual transfer, cross-task—and more broadly, universal cross-task and cross-lingual—generalization, as explored in the Polyglot Prompt paper (Fu et al., 2022a), remains an open direction. These design choices reflect the practical scope of this study and define a foundation for broader future extensions.

## Acknowledgements

This work was partially supported by the European Union under Horizon Europe project "GAIN" (GA #101078950) and by the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the project TRAILS (01IW24005).

## 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.

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

Philipp Borchert, Ivan Vulić, Marie-Francine Moens, and Jochen De Weerdt. 2025. **Language fusion for parameter-efficient cross-lingual transfer**. *Preprint*, arXiv:2501.06892.

Joon-Young Choi, Junho Kim, Jun-Hyung Park, Wing-Lam Mok, and SangKeun Lee. 2023. **SMoP: Towards efficient and effective prompt tuning with sparse mixture-of-prompts**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14306–14316, Singapore. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, and Pengfei Liu. 2022a. **Polyglot prompt: Multilingual multitask prompt training**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9919–9935, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, and Pengfei Liu. 2022b. Polyglot prompt: Multilingual multitask prompt training. *EMNLP*.

Katharina Häggerl, Jindřich Libovický, and Alexander Fraser. 2024. **Understanding cross-lingual Alignment—A survey**. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10922–10943, Bangkok, Thailand. Association for Computational Linguistics.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. **Parameter-efficient fine-tuning for large models: A comprehensive survey**. *Preprint*, arXiv:2403.14608.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. **Lora: Low-rank adaptation of large language models**. *Preprint*, arXiv:2106.09685.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. **Xtreme: a massively multilingual multi-task benchmark for evaluating cross-lingual generalization**. In *Proceedings of the 37th International Conference on Machine Learning*, ICML’20. JMLR.org.

Lianzhe Huang, Shuming Ma, Dongdong Zhang, Furu Wei, and Houfeng Wang. 2022. **Zero-shot cross-lingual transfer of prompt-based tuning with a unified multilingual prompt**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11488–11497, Abu Dhabi, United Arab Emirates. 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.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. **From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

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.

Shuang Li, Xuming Hu, Aiwei Liu, Yawen Yang, Fukun Ma, Philip S. Yu, and Lijie Wen. 2023. **Enhancing cross-lingual natural language inference by soft prompting with multilingual verbalizer**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1361–1374, Toronto, Canada. 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.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Motta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

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

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

Fred Philippy, Siwen Guo, 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.

Xiaoyu Qiu, Yuechen Wang, Jiaxin Shi, Wengang Zhou, and Houqiang Li. 2024. Cross-lingual transfer for natural language inference via multilingual prompt translator. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Anastasiia Razdaibiedina, Yuning Mao, Madian Khabsa, Mike Lewis, Rui Hou, Jimmy Ba, and Amjad Almalhairi. 2023. Residual prompt tuning: improving prompt tuning with residual reparameterization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6740–6757, Toronto, Canada. Association for Computational Linguistics.

Sona Elza Simon and Preethi Jyothi. 2025. Deftx: De-noised sparse fine-tuning for zero-shot cross-lingual transfer. *Preprint*, arXiv:2505.15090.

Luping Wang, Sheng Chen, Linnan Jiang, Shu Pan, Runze Cai, Sen Yang, and Fei Yang. 2025. Parameter-efficient fine-tuning in large models: A survey of methodologies. *Preprint*, arXiv:2410.19878.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogério Feris, Huan Sun, and Yoon Kim. 2023. Multitask prompt tuning enables parameter-efficient transfer learning. *Preprint*, arXiv:2303.02861.

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

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

Bo Zheng, Li Dong, Shaohan Huang, Wenhui Wang, Zewen Chi, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. Consistency regularization for cross-lingual fine-tuning. 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 3403–3417, Online. Association for Computational Linguistics.

## A ZS-XLT Per-Language Results

Language	Group	EnArZho				Joshi5				Seen			
		SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE	SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE	SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE
ace_Arab	2	36.5	34.9	34.4	32.4	36.1	35.6	35.9	33.6	36.5	35.5	37.7	37.2
aka_Latn	2	42.3	44.5	45.0	41.8	44.3	43.8	45.9	47.3	48.0	48.4	50.4	52.9
arb_Latn	2	36.7	37.3	38.4	35.7	35.5	38.0	39.7	41.7	39.9	43.7	44.3	48.0
ayr_Latn	2	41.0	41.2	40.5	42.6	41.7	42.9	42.1	45.8	45.5	47.0	47.6	46.1
bam_Latn	2	30.6	34.9	36.9	33.8	32.5	35.0	34.1	41.3	36.4	38.0	38.7	43.3
bem_Latn	2	45.1	45.0	45.1	41.1	45.8	46.4	47.6	47.0	52.7	52.8	53.1	52.8
bjn_Arab	2	26.5	26.8	27.2	25.2	28.4	27.5	24.6	27.1	27.6	27.0	27.8	27.5
bod_Tibt	2	22.6	25.7	24.6	25.4	24.5	24.1	24.4	25.8	25.6	24.7	26.1	26.3
cjk_Latn	2	45.7	44.2	42.5	41.5	43.9	45.0	44.7	45.2	46.4	45.2	46.3	48.1
ckb_Arab	2	28.4	29.6	30.6	25.0	29.5	33.8	31.0	29.5	36.7	35.2	37.0	38.2
dyu_Latn	2	40.7	40.9	40.9	41.4	41.6	42.4	44.1	46.2	47.1	49.0	48.7	49.3
dzo_Tibt	2	20.2	24.6	23.3	24.5	24.1	22.6	23.7	24.2	24.7	22.9	23.8	24.2
ewe_Latn	2	34.8	36.1	36.7	39.1	34.7	36.8	37.8	43.3	40.9	40.8	42.9	41.9
fon_Latn	2	31.1	33.7	35.0	35.4	33.3	33.9	36.4	41.6	37.2	36.7	39.2	43.1
ibo_Latn	2	37.9	38.5	37.9	38.3	38.6	38.0	39.3	42.4	40.0	37.7	40.8	43.4
kab_Latn	2	26.9	29.6	27.5	31.3	26.6	27.7	28.6	32.8	30.3	29.6	32.2	33.2
kam_Latn	2	43.5	43.8	42.6	40.6	42.0	44.8	43.9	45.9	46.4	46.0	48.2	48.6
kbp_Latn	2	29.0	33.7	32.0	34.4	31.3	33.7	32.2	39.2	36.5	36.3	35.4	43.3
kik_Latn	2	40.2	43.1	43.4	43.6	42.3	43.2	43.8	45.1	46.7	47.3	47.2	48.5
kin_Latn	2	40.1	39.0	37.5	39.0	39.4	40.7	39.9	45.5	44.2	43.9	45.4	46.3
kmb_Latn	2	40.3	42.1	41.5	40.2	41.2	43.4	43.2	44.5	43.7	45.3	46.4	48.6
knc_Arab	2	25.0	24.6	25.6	24.5	25.6	25.9	23.8	27.2	25.1	23.8	24.5	26.2
lua_Latn	2	44.1	45.8	47.3	47.2	45.8	47.0	47.4	51.8	46.7	48.7	48.9	53.0
lug_Latn	2	34.9	35.9	36.5	35.3	37.5	37.7	38.3	40.2	42.1	42.3	43.4	45.2
min_Arab	2	29.0	28.7	27.3	26.4	28.5	29.3	27.2	30.0	30.4	29.0	31.1	30.6
mmi_Beng	2	32.5	34.5	33.3	30.0	33.6	34.0	35.2	31.4	38.0	36.5	35.6	37.6
mos_Latn	2	38.5	40.4	40.0	38.3	39.8	42.0	42.9	44.6	41.0	44.0	44.9	46.9
mri_Latn	2	36.7	38.9	37.9	39.2	36.0	38.4	38.8	44.1	35.1	38.8	39.2	44.6
nqo_Nkoo	2	23.7	24.4	25.5	23.6	24.6	22.0	24.1	23.5	24.3	21.9	24.5	24.8
nso_Latn	2	41.4	42.7	41.4	41.0	42.4	42.7	45.8	46.7	46.0	48.3	46.9	48.2
nus_Latn	2	25.7	27.5	28.1	28.6	28.0	28.0	27.4	31.7	30.4	28.9	31.4	33.6
run_Latn	2	38.0	41.1	39.7	39.8	38.9	40.9	41.4	44.0	44.1	44.4	46.4	46.9
sat_Olck	2	22.6	25.0	25.2	23.4	24.3	22.4	23.9	23.1	23.0	20.8	22.2	22.1
shn_Mymr	2	34.0	35.5	35.4	30.8	34.9	35.4	35.1	35.5	38.6	39.9	40.8	42.6
smo_Latn	2	40.1	41.7	41.7	38.2	38.6	39.3	43.5	41.4	41.4	42.6	43.7	47.5
sna_Latn	2	41.3	42.9	41.4	39.2	42.3	43.2	42.7	43.9	46.8	46.8	46.8	49.2
sot_Latn	2	40.7	39.8	39.6	39.6	41.2	42.7	43.1	46.8	45.5	47.5	47.5	49.4
ssw_Latn	2	45.7	45.3	45.8	39.6	45.9	48.1	49.2	45.1	52.1	53.0	52.5	52.8
taq_Latn	2	40.7	42.5	41.4	41.7	40.2	42.4	43.3	44.5	44.5	45.6	46.5	48.1
taq_Tfng	2	26.7	27.6	28.2	27.3	27.0	24.4	27.0	26.1	27.1	25.6	25.3	28.7
tgk_Cyrl	2	39.4	40.8	39.1	42.2	40.3	42.7	43.2	44.5	44.9	47.2	48.6	51.2
tsn_Latn	2	36.8	39.2	38.5	39.0	39.8	41.1	42.7	46.0	43.4	44.2	45.7	47.5
tso_Latn	2	39.2	40.1	38.8	37.7	38.7	41.9	41.9	44.3	45.4	44.6	46.5	45.2
tzm_Tfng	2	24.3	25.5	25.4	23.6	25.7	22.6	24.2	24.3	25.0	23.4	24.8	25.1
umb_Latn	2	39.6	39.8	41.2	38.8	39.0	40.8	42.2	42.8	42.7	43.2	46.5	47.1
yor_Latn	2	39.0	40.1	38.5	37.6	40.1	40.0	40.9	39.9	41.1	41.7	41.4	42.9
ace_Latn	1	64.7	65.8	64.7	62.4	66.0	67.0	67.7	67.5	72.8	74.0	74.8	74.3
acm_Arab	1	87.4	87.5	87.3	86.7	88.1	87.5	90.0	88.0	88.7	89.8	89.6	89.2
acq_Arab	1	88.2	87.9	88.7	87.6	89.4	88.7	90.4	87.9	90.0	90.0	90.5	88.8
aeb_Arab	1	85.1	85.6	85.5	84.5	87.5	86.7	87.5	85.7	87.9	87.3	87.7	88.3
ajp_Arab	1	87.0	87.7	87.7	87.3	88.1	87.5	88.6	87.9	89.3	88.5	89.3	89.2
apc_Arab	1	87.7	88.8	88.9	89.0	88.2	88.7	90.2	89.3	89.8	90.7	90.3	89.7
ars_Arab	1	88.9	88.4	88.6	88.1	89.4	88.8	90.9	88.2	90.6	91.0	91.3	89.5
ary_Arab	1	84.7	85.7	85.2	85.0	86.1	85.8	86.8	86.3	86.7	86.7	86.7	87.5
arz_Arab	1	86.8	86.9	86.3	87.4	88.7	86.3	89.2	87.5	89.3	89.2	89.2	88.6
ast_Latn	1	85.7	85.5	85.3	84.3	85.2	85.8	86.6	86.0	85.9	87.3	87.0	87.8
awa_Deva	1	85.1	85.4	85.3	84.8	84.1	85.0	87.0	85.2	86.4	87.6	87.8	87.4
bak_Cyrl	1	64.9	66.9	66.1	62.9	66.5	65.8	69.1	67.2	71.0	71.9	72.6	72.3
ban_Latn	1	79.2	78.4	77.6	75.2	80.5	79.3	82.2	77.8	82.2	83.3	83.9	84.3
bho_Deva	1	82.1	82.6	82.3	80.3	82.6	81.5	82.5	79.7	83.3	83.6	83.4	82.9
bjn_Latn	1	76.4	75.3	75.5	72.7	78.1	76.4	79.3	75.3	79.7	80.5	81.6	79.8
bug_Latn	1	61.6	63.3	61.2	62.1	62.4	62.7	65.5	67.2	67.0	68.7	69.4	69.4
ceb_Latn	1	74.4	74.2	74.1	69.5	76.7	74.7	75.5	74.6	79.2	79.1	78.0	77.7
crh_Latn	1	81.5	82.5	80.2	77.1	82.4	81.6	82.2	80.1	83.4	83.8	84.0	84.6
dik_Latn	1	39.3	41.4	41.8	40.9	41.5	41.0	44.4	45.6	45.6	47.7	48.6	49.0
fao_Latn	1	77.7	79.8	78.2	73.9	80.0	79.7	81.9	76.3	81.6	83.6	82.8	81.8
fij_Latn	1	40.7	40.7	40.0	38.1	42.5	39.9	41.1	43.5	42.0	42.1	43.7	45.5

Language	Group	EnArZho				Joshi5				Seen			
		SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE	SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE	SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE
fur_Latn	1	71.7	73.9	73.3	67.0	73.4	73.7	77.2	71.8	77.7	81.4	80.9	80.4
fuv_Latn	1	46.4	46.7	46.0	44.0	48.1	48.5	48.7	48.4	51.7	51.7	52.8	53.6
grn_Latn	1	61.1	64.6	64.4	65.4	62.2	63.7	66.2	67.6	65.9	66.6	69.0	70.6
hat_Latn	1	54.7	57.2	55.9	50.7	57.5	57.7	58.7	56.9	62.6	62.7	63.5	62.8
hne_Deva	1	83.5	85.0	84.1	82.5	84.7	84.6	85.8	83.3	86.4	87.0	87.2	86.6
ilo_Latn	1	63.2	64.3	64.0	63.0	64.8	65.0	67.6	67.1	68.2	69.8	70.0	70.9
kac_Latn	1	38.2	39.5	40.1	37.8	38.2	42.0	41.8	43.5	41.4	43.2	45.1	44.0
kas_Arab	1	68.7	69.1	69.2	68.1	68.4	69.1	71.3	68.8	71.2	72.0	72.8	73.6
kas_Deva	1	61.5	63.0	61.5	58.5	62.2	63.6	64.0	59.5	65.2	68.3	67.8	69.0
kea_Latn	1	74.5	75.2	74.7	72.8	76.0	77.0	77.9	74.8	80.2	80.2	80.2	80.1
knc_Latn	1	43.7	45.4	43.4	44.5	45.9	46.3	47.3	48.4	52.2	52.5	51.9	50.2
kon_Latn	1	49.3	51.2	50.6	47.0	51.0	52.5	51.2	53.4	51.7	53.8	55.6	54.2
lij_Latn	1	71.3	73.3	73.5	68.9	74.0	73.6	76.1	72.0	76.6	77.1	77.8	79.0
lim_Latn	1	74.7	77.0	75.4	71.4	78.4	76.6	79.2	76.9	81.3	82.5	83.5	82.5
lin_Latn	1	49.3	49.3	48.4	46.7	50.3	49.2	51.0	50.6	51.7	51.6	52.5	52.2
lmo_Latn	1	71.7	74.1	73.1	64.9	74.5	75.4	76.3	71.4	77.7	79.9	79.4	79.9
ltg_Latn	1	73.1	74.3	72.0	68.4	74.9	72.7	75.8	72.3	76.5	77.2	77.3	78.1
ltz_Latn	1	67.0	70.3	69.7	65.6	68.7	69.6	72.7	69.5	73.7	74.6	75.0	74.9
luo_Latn	1	42.0	42.6	40.5	41.3	41.5	43.0	44.1	44.3	46.9	46.8	47.6	47.3
lus_Latn	1	58.1	58.7	59.3	56.2	59.6	59.2	61.9	60.0	64.6	66.6	65.5	65.3
mag_Deva	1	83.8	84.6	84.7	82.8	84.4	83.9	85.8	83.7	86.5	87.5	87.0	87.5
mai_Deva	1	84.3	84.5	85.1	83.9	84.9	84.3	86.8	83.7	86.8	86.9	86.6	86.7
min_Latn	1	80.0	79.3	78.4	73.6	82.1	81.7	82.4	79.4	84.9	86.1	86.1	85.6
mlt_Latn	1	57.2	60.2	58.4	55.8	60.0	60.1	62.9	61.5	65.9	67.8	66.4	68.9
nya_Latn	1	47.8	48.3	48.7	47.0	48.4	51.3	51.8	50.7	52.0	54.6	54.2	56.2
oci_Latn	1	82.1	84.8	84.0	81.2	84.1	83.9	86.0	84.4	86.1	86.2	86.5	87.3
pag_Latn	1	69.5	70.2	70.0	68.8	70.4	70.1	70.9	71.1	71.3	73.3	72.7	73.0
pap_Latn	1	73.8	76.0	75.7	71.7	74.4	75.7	76.6	75.1	76.8	77.5	78.1	77.5
prs_Arab	1	89.3	89.2	88.8	87.9	88.8	89.1	90.0	87.8	90.2	89.9	90.0	89.5
quy_Latn	1	49.4	51.4	50.5	51.4	51.2	50.4	54.1	54.6	54.4	56.7	57.4	58.2
sag_Latn	1	42.7	46.6	44.2	40.6	42.8	45.1	47.2	47.4	46.1	47.2	47.8	48.4
scn_Latn	1	68.3	70.8	70.1	66.1	70.4	71.2	73.6	69.8	73.3	74.7	75.0	75.6
srd_Latn	1	72.3	74.5	72.4	66.3	74.5	74.1	75.8	70.2	77.7	78.0	78.0	77.5
szl_Latn	1	76.8	78.4	77.0	72.4	77.9	77.0	80.4	76.4	81.6	82.4	82.8	83.7
tat_Cyrl	1	69.0	70.7	68.9	64.8	71.5	71.2	72.4	70.1	75.0	76.5	75.8	75.4
tgl_Latn	1	83.2	83.7	83.3	82.6	85.0	84.8	85.7	86.0	86.1	86.2	87.1	87.5
tir_Ethi	1	52.6	56.8	58.4	49.3	53.6	57.3	58.5	54.4	59.8	63.9	62.9	62.5
tpi_Latn	1	67.2	68.2	68.5	62.5	69.0	69.1	71.9	66.7	70.9	70.6	71.6	73.8
tuk_Latn	1	59.8	59.0	56.5	53.9	61.9	60.2	61.7	55.6	65.3	65.6	65.4	66.3
tum_Latn	1	42.6	43.5	44.3	40.9	43.5	48.2	46.6	47.8	49.1	49.0	50.2	51.5
twi_Latn	1	42.0	45.8	46.7	43.5	44.8	44.9	48.2	49.0	48.8	51.2	52.8	55.8
vec_Latn	1	76.4	80.0	78.8	73.7	79.4	79.2	81.9	77.1	82.2	84.2	84.4	82.9
war_Latn	1	75.5	74.9	74.8	72.1	77.4	76.8	78.3	76.6	79.9	81.3	80.9	80.9
wol_Latn	1	48.3	48.5	47.7	50.7	49.7	51.1	50.2	53.7	56.7	56.1	55.5	57.2
yue_Hant	1	87.7	87.4	87.0	88.7	87.7	88.0	88.8	87.0	87.5	89.0	89.3	87.5
zul_Latn	1	58.8	60.8	60.0	51.6	60.9	61.7	64.8	56.7	66.2	67.2	67.7	67.5
afr_Latn	0	87.2	87.5	87.7	86.5	87.8	87.8	89.4	88.2				
als_Latn	0	86.7	87.2	86.9	86.4	87.9	87.9	89.2	87.8				
amh_Ethi	0	80.9	81.7	81.1	79.6	82.2	81.3	82.4	81.1				
asm_Beng	0	80.8	82.5	83.4	74.8	82.5	83.7	86.3	80.0				
azb_Arab	0	74.7	73.8	73.1	72.8	75.2	73.9	75.8	75.2				
azj_Latn	0	87.3	87.7	87.0	87.9	87.6	87.5	88.8	89.3				
bel_Cyrl	0	86.3	87.5	87.5	86.9	88.2	87.8	89.6	86.6				
ben_Beng	0	84.7	84.4	84.4	82.7	84.7	84.8	86.2	85.8				
bos_Latn	0	88.2	87.9	88.7	87.7	88.8	88.8	90.3	89.0				
bul_Cyrl	0	87.5	87.4	88.0	88.8	89.2	88.7	89.0	88.6				
cat_Latn	0	89.0	88.4	88.6	87.2	89.2	89.0	90.3	88.1				
ces_Latn	0	89.6	89.0	89.8	88.9	90.5	89.7	90.9	89.6				
cym_Latn	0	83.8	83.9	82.9	78.7	84.5	85.0	86.2	82.9				
dan_Latn	0	88.3	88.7	89.0	88.3	89.5	89.0	90.7	89.2				
ell_Grek	0	86.2	88.2	87.7	85.4	88.2	88.1	89.2	87.1				
epo_Latn	0	88.4	87.5	87.3	85.7	89.0	87.9	89.3	86.4				
est_Latn	0	89.1	87.5	87.7	84.2	89.1	88.6	89.7	88.6				
eus_Latn	0	86.7	85.8	85.9	84.2	88.4	86.8	89.3	85.7				
fin_Latn	0	87.5	87.8	87.1	85.9	88.7	88.2	89.3	87.7				
gaz_Latn	0	47.9	49.3	46.7	39.6	50.2	51.7	51.8	46.6				
gla_Latn	0	75.8	76.5	75.6	59.9	77.1	77.4	78.7	69.6				
gle_Latn	0	82.3	83.3	80.9	73.1	83.3	84.1	85.2	79.8				
glg_Latn	0	88.7	88.3	87.8	88.2	88.5	88.7	88.9	88.3				
guj_Gujr	0	84.4	84.8	84.2	82.5	84.6	84.2	86.2	84.5				

Language	Group	EnArZho				Joshi5				Seen			
		SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE	SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE	SPT	D <sup>XPE-30</sup>	D <sup>XPE-70</sup>	XPE
hau_Latn	0	76.7	76.6	72.3	64.1	78.5	78.4	79.7	73.5				
heb_Hebr	0	86.2	85.9	85.9	85.0	86.3	86.5	87.1	85.9				
hin_Deva	0	87.7	87.6	87.7	85.7	87.7	87.3	88.3	87.4				
hrv_Latn	0	89.6	89.5	89.5	89.2	90.2	89.9	91.1	89.2				
hun_Latn	0	87.9	87.7	88.2	87.9	88.6	88.8	88.7	87.7				
hye_Armn	0	86.8	86.9	86.8	84.0	87.1	86.8	88.6	86.8				
ind_Latn	0	90.1	90.1	90.3	89.2	90.7	90.0	91.2	90.3				
isl_Latn	0	87.3	86.6	86.8	86.9	88.8	87.6	89.3	88.9				
ita_Latn	0	89.1	88.9	89.5	88.8	89.3	90.3	90.4	89.6				
jav_Latn	0	83.4	83.3	82.3	80.2	84.3	84.2	84.1	82.2				
kan_Knda	0	85.5	85.6	87.0	83.3	86.9	86.7	89.0	86.6				
kat_Geor	0	87.5	88.5	88.4	86.8	88.5	88.4	89.5	89.5				
kaz_Cyril	0	86.8	87.4	87.0	85.8	87.9	87.6	87.8	87.9				
khk_Cyril	0	83.4	84.1	84.6	83.2	85.0	84.0	86.1	84.8				
khm_Khmr	0	84.5	86.2	85.6	84.5	85.6	85.5	87.3	86.6				
kir_Cyril	0	85.6	86.1	86.1	87.4	87.6	85.9	88.5	87.5				
kmr_Latn	0	76.3	76.9	76.8	69.4	77.8	77.3	78.9	74.6				
kor_Hang	0	86.6	86.9	86.1	87.1	87.2	86.8	88.0	88.2				
lao_Laoo	0	85.1	86.9	86.7	86.6	86.6	85.6	88.8	88.7				
lit_Latn	0	86.9	86.5	87.1	85.5	86.8	87.0	89.4	88.3				
lvs_Latn	0	88.5	87.3	87.5	87.8	88.3	89.2	89.8	87.7				
mal_Mlym	0	85.4	86.6	85.2	82.5	87.5	85.6	86.9	85.6				
mar_Deva	0	87.4	87.0	85.8	82.4	87.4	86.3	88.1	85.1				
mkd_Cyril	0	85.9	86.5	86.3	85.4	86.8	86.9	88.1	85.9				
mya_Mymr	0	83.5	85.1	85.0	83.1	85.3	84.9	86.5	84.5				
nld_Latn	0	88.9	89.2	89.2	87.7	89.2	89.4	90.0	89.5				
nno_Latn	0	88.6	88.8	88.2	86.8	88.9	88.7	90.5	88.8				
nob_Latn	0	88.1	87.1	87.2	87.5	88.1	87.6	88.6	87.8				
npi_Deva	0	86.2	86.1	86.6	84.4	86.7	87.3	87.5	85.2				
ory_Orya	0	81.8	81.4	81.8	79.5	83.0	83.4	84.1	80.8				
pan_Guru	0	83.6	83.2	84.2	80.8	84.1	83.3	85.6	83.1				
pbt_Arab	0	82.0	81.8	81.6	79.6	82.2	82.3	84.2	80.8				
pes_Arab	0	89.9	89.7	89.6	90.1	89.9	89.7	90.5	89.8				
plt_Latn	0	74.6	75.8	73.4	66.8	77.8	77.0	78.5	71.1				
pol_Latn	0	88.7	88.7	89.2	88.0	90.0	89.1	89.9	87.7				
por_Latn	0	88.5	88.6	87.7	87.5	89.1	88.4	89.8	87.3				
ron_Latn	0	87.9	87.6	87.6	87.0	88.9	89.0	89.7	87.5				
rus_Cyril	0	87.8	87.7	88.0	88.7	88.9	88.0	88.9	87.9				
san_Deva	0	81.7	81.8	81.2	78.7	81.8	81.3	83.7	80.9				
sin_Sinh	0	83.4	83.8	83.4	81.7	84.8	83.8	86.0	82.6				
slk_Latn	0	88.6	87.8	88.1	88.4	89.0	88.7	89.7	89.4				
slv_Latn	0	86.6	85.2	86.1	85.8	87.4	87.1	88.3	85.9				
snd_Arab	0	83.3	82.9	83.3	81.4	83.7	82.8	84.3	84.1				
som_Latn	0	72.5	74.1	72.2	67.7	74.3	74.1	75.6	71.1				
srp_Cyril	0	87.7	87.7	88.8	87.3	89.0	87.9	89.9	89.0				
sun_Latn	0	84.4	84.3	84.9	81.3	86.4	85.0	87.0	83.9				
swe_Latn	0	88.1	88.7	88.6	87.2	88.8	88.5	89.4	88.4				
swi_Latn	0	79.6	80.3	79.9	79.1	81.2	81.2	83.6	81.6				
tam_Taml	0	84.8	84.5	84.5	83.3	85.3	85.6	86.4	85.4				
tel_Telu	0	85.7	86.0	87.1	85.2	86.2	86.0	87.9	86.9				
tha_Thai	0	87.6	87.7	87.8	89.8	89.1	88.2	89.8	89.1				
tur_Latn	0	88.0	87.7	88.5	87.5	88.9	88.4	89.3	87.5				
uig_Arab	0	81.1	81.8	81.5	79.1	82.8	81.4	82.5	80.5				
ukr_Cyril	0	90.0	88.9	89.8	89.3	90.7	90.7	91.2	90.3				
urd_Arab	0	86.2	86.3	86.3	86.6	87.0	86.2	88.0	87.5				
uzn_Latn	0	84.3	84.1	83.0	81.3	85.0	84.7	85.5	83.1				
vie_Latn	0	89.8	89.7	89.2	90.4	90.1	90.2	90.1	90.0				
xho_Latn	0	60.4	62.5	62.4	51.5	62.5	62.9	65.1	58.5				
ydd_Hebr	0	76.3	73.1	71.7	65.6	75.7	76.6	76.6	69.8				
zho_Hant	0	89.9	89.3	89.0	89.2	89.5	90.0	90.3	88.7				
zsm_Latn	0	90.5	89.4	89.2	88.3	90.3	89.8	91.5	89.9				

**ZS-XLT performance (accuracy) per target language.** The "Language" column corresponds to language codes in SIB-200 dataset. While all listed languages collectively form the *All /wo Joshi5* group, the "Group" column indicates which sub-target group(s) each language belongs to: 2 = *Low-Performing* and *Unseen*, 1 = *Unseen*, 0 = *Seen /wo Joshi5*. Each set of four method columns corresponds to a different source language group. D<sup>XPE-30</sup> and D<sup>XPE-30</sup> and D<sup>XPE-70</sup> are abbreviated forms of the DUAL setup.