Leveraging Routing Dynamics in Mixture-of-Experts Models for Efficient Language Adaptation

Aditi Khandelwal **Marius Mosbach** **Verna Dankers** **Siva Reddy** **Golnoosh Farnadi**

Mila – Quebec AI Institute & McGill University

{aditi.khandelwal, marius.mosbach, verna.dankers, siva.reddy, gfarnadi}@mila.quebec

Abstract

Mixture-of-Experts (MoE) models are widely used to scale language models, yet their expert routing behavior and adaptation in a multilingual setting remain underexplored. In this work, we study multilingual routing dynamics during continual pre-training of an English-centric MoE model on a multilingual corpus, analyzing how expert usage varies across languages. We find that continual multilingual pre-training leads to diffused, language-agnostic routing in early and middle layers, with language specialization primarily emerging in the final layers. We also show that token-level vocabulary overlap between languages plays an important role in how languages are routed. Motivated by these findings, we propose a parameter-efficient adaptation strategy that updates language-specific and shared experts in the final MoE layers. Experiments on Multi-BLiMP and Belebele show that our method achieves a strong performance–efficiency trade-off, attaining competitive performance relative to fine-tuning complete final layers, while updating less than 2% of the parameters. Overall, our findings provide insights into where and how language specialization emerges in MoEs during continual pre-training and provide practical insights for low-resource multilingual adaptation. Our code is available here.

1 Introduction

Mixture-of-Experts (MoE) architectures are now the standard for scaling *Large Language Models* (LLMs) because they allow for massive parameter counts while maintaining manageable inference costs (DeepSeek-AI et al., 2025; Comanici et al., 2025; Team et al., 2025). By sparsely activating a subset of the parameters per token, MoE models offer both computational efficiency and a modular structure that makes internal routing behavior amenable to analysis (Xue et al., 2024; Lo et al., 2025; Li et al., 2025). While this modularity has been shown to facilitate specialization in domains such as math or coding (Li et al.; Jiang et al., 2024; Muennighoff et al.), the extent to which MoEs develop specialized routing strategies for different languages remains underexplored. At the same time, adapting large MoEs to new languages is computationally demanding. Although their modularity should, in principle, enable parameter-efficient adaptation by updating only a subset of experts, it remains unclear which experts to update without first understanding multilingual routing behavior.

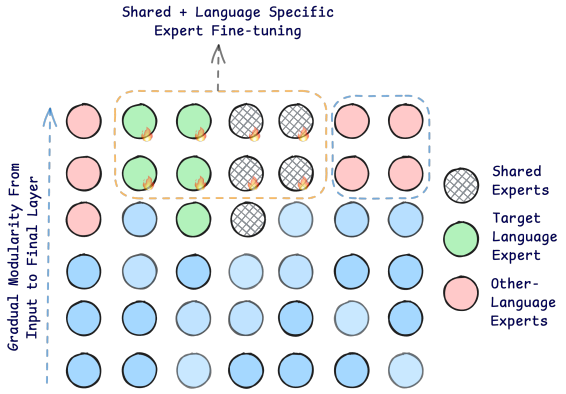


Figure 1: We hypothesize that continually pre-training English-centric MoEs on a mixture of high-resource languages leads to the emergence of language experts. We leverage this specialization for parameter-efficient adaptation to related low-resource languages.

Prior research has explored the internal mechanisms of dense multilingual models. Findings suggest that they often rely on English as an internal 'pivot' or 'concept space' in the middle layers (Alabi et al., 2024; Wendler et al., 2024), with language-specific neurons primarily localized in the first and last few layers (Kojima et al., 2024). In contrast, the routing dynamics of multilingual MoE models are less well understood. Recent work has begun to address this gap, observing that MoEs exhibit language-agnostic routing in intermediate layers and specialization in early and late layers (Bandarkar et al., 2026). However, these analy-

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 54675–54690
October 24-29, 2026 ©2026 Association for Computational Linguistics

ses are derived predominantly from models trained on English-dominant corpora (Abdin et al., 2024b; Muennighoff et al.; Agarwal et al., 2025).

We ask: *does continual pre-training of an MoE model on balanced multilingual data lead to language-specific specialization, and if so, can this behavior be leveraged to efficiently adapt a model to related low-resource languages?*

We conduct a controlled study of the routing dynamics of the OLMoE model (Muennighoff et al.) during multilingual continual pre-training. We adapt OLMoE-Base on a 35B-token corpus spanning seven languages. Our analysis reveals that routing becomes diffuse and language-agnostic in early and middle layers, with distinct language specialization emerging only in the final layers.

Leveraging these insights, we investigate whether such language specialization can be used practically for adaptation to low-resource languages. We introduce *Selective and Shared Expert Finetuning* (SSFT), which updates only the relevant language-specific experts and a small set of shared experts in the final layers. Our results demonstrate that selectively finetuning both specialized and shared experts provides the best trade-off between parameter efficiency and performance. This hybrid strategy consistently outperforms the adaptation of specialized experts alone and achieves strong performance while updating a substantially smaller fraction of parameters ($< 2\%$ of the model).

Our core contributions are: 1) **Multilingual Continual Pre-training Analysis.** With a controlled study of routing dynamics under balanced multilingual training, we show diffused and language-agnostic routing in early to middle layers, with language specialization in final layers. We identify *vocabulary overlap* between languages as an important factor influencing routing behavior.

2) **Selective and Shared Expert Finetuning (SSFT).** We propose SSFT, showing that both language-specific and shared experts are important for low-resource adaptation. This strategy offers a favorable performance–efficiency trade-off.

The rest of this paper is structured as follows: §2 provides background on MoE architectures and the metrics we use to analyze routing dynamics. §3 describes our balanced multilingual continual pre-training setup and examines the evolution of routing dynamics. §4 proposes and analyzes parameter-efficient adaptation methods. We end by discussing related work in §5 and conclude in §6.

2 Preliminaries

Below, we provide the necessary background on the MoE architecture and the OLMoE model we use, followed by the mathematical framework employed to analyze the routing dynamics.

2.1 Mixture-of-Experts

MoE models extend transformer LMs by replacing the standard feed-forward block with a collection of expert feed-forward networks (Shazeer et al., 2017; Lepikhin et al.; Fedus et al., 2022). At each MoE layer, a trainable router (a small linear projection followed by a softmax) assigns routing probabilities across all experts. For each input token, only the top-k experts with the highest routing probabilities are selected to process it. The outputs of the chosen experts are then weighted by their routing probabilities and summed, producing the token representation passed to the next layer.

We base our experiments on OLMoE-Base[1], a decoder-only transformer with MoE feed-forward layers. Each MoE layer consists of 64 experts and a learned router that selects the top 8 experts per token in each layer. Experts are identical in architecture but maintain separate parameters, enabling modularity through routing. OLMoE-Base (Muennighoff et al.) has been pretrained on approximately 5T predominantly English tokens using a data mixture of the DCLM-Baseline corpus (Li et al., 2024) and Dolma 1.7 (Soldaini et al., 2024).

2.2 Analyzing routing behavior

To study multilingual processing in MoE models, we analyze expert routing patterns across languages and layers to characterize how routing mass is distributed. The methods introduced here form the basis for the routing analyses presented in §3.

We collect routing information on held-out documents for each language across all decoder layers. For a given language ℓ and layer k, we record post-softmax routing probabilities for all tokens.

Let E denote the number of experts in each MoE layer. For the i-th document from language ℓ, let T_i denote its number of tokens. We denote by $\mathbf{p}_{i,t}^{(k)}(\ell) \in \Delta^{E-1}$ the routing probability distribution for the t-th token of this document at layer k.[2] For each document, we compute a document-

[1]We use OLMoE due to its fully open-source implementation and publicly available training code, which enables reproducible analysis of routing and expert specialization.

[2]This vector contains one element per expert, representing the probability of routing this token to that expert.

level expert usage distribution by averaging routing probabilities across tokens:

$$\mathbf{q}_i^{(k)}(\ell) = \frac{1}{T_i} \sum_{t=1}^{T_i} \mathbf{p}_{i,t}^{(k)}(\ell) \, .$$

This quantity represents the average fraction of routing mass assigned to each expert for a single document, summarizing how that document is routed through the MoE layer.

We then aggregate document-level distributions to obtain a language-level expert usage distribution for each layer. Let N_ℓ denote the number of documents for language ℓ; the aggregated distribution is given by $\mathbf{q}^{(k)}(\ell) = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \mathbf{q}_i^{(k)}(\ell)$. This distribution captures the typical expert usage pattern for language ℓ at layer k, averaged across documents.

Router entropy. To quantify how concentrated expert routing is for a given language, we compute the router entropy at each layer:

$$H_k(\ell) = -\sum_{e=1}^{E} \mathbf{q}^{(k)}(\ell)[e] \, \log \mathbf{q}^{(k)}(\ell)[e] \, .$$

Here, $\mathbf{q}^{(k)}(\ell)[e]$ denotes the e-th element of the vector $\mathbf{q}^{(k)}(\ell)$, representing the average routing probability to expert e. Lower entropy indicates that tokens from language ℓ are routed to a small subset of experts, while higher entropy reflects more diffuse routing across experts.

Cross-lingual routing divergence. To compare routing behavior between languages, we compute the Jensen–Shannon divergence between their language-level expert usage distributions. For two languages ℓ_i and ℓ_j at layer k, we define

$$\mathrm{JSD}_k(\ell_i, \ell_j) = \mathrm{JSD}\Big(\mathbf{q}^{(k)}(\ell_i), \, \mathbf{q}^{(k)}(\ell_j)\Big) \, .$$

Low JSD values indicate similar expert usage patterns between the two languages, while higher values indicate more distinct routing behavior.

2.3 Language selection

We use a high- and a low-resource language group for our analyses. The high-resource set contains English (en), Arabic (ar), Czech (cs), Spanish (es), Finnish (fi), Hindi (hi), and Russian (ru). We use this set in §3 to analyze how expert routing evolves during continual pre-training and how these patterns differ from English-centric models.

The low-resource set contains Catalan (ca), Estonian (et), Marathi (mr), Slovak (sk), Ukrainian (uk), Dutch (nl) and Urdu (ur).[3] We use this set in §3 to analyze cross-lingual co-routing patterns alongside the high-resource set, and in §4 to evaluate different adaptation strategies. Each low-resource language is paired with a high-resource anchor from the same family, yielding Catalan–Spanish, Estonian–Finnish, Marathi–Hindi, Slovak–Czech, Ukrainian–Russian, and Urdu–Arabic, paired based on token-level vocabulary overlap, script similarity, and the availability of downstream evaluation benchmarks.[4][5] The resulting pairs span a wide range of overlap, from Spanish–Catalan at approximately 20% to Hindi–Marathi at over 90%.

3 Routing Dynamics during Continual Pre-training

In a multilingual setting, MoEs potentially allow for experts specializing in individual languages. However, their internal language-specific or language-agnostic routing behavior remains understudied. Here, we analyze MoE expert routing during multilingual continual pre-training to assess the emergence of language-specific experts, before leveraging these insights for model adaptation (§4).

Hypothesis. When an English-centric MoE model is continually pre-trained on balanced multilingual data, expert routing reorganizes to become increasingly language-sensitive, reducing English-dominated routing patterns and exhibiting greater differentiation across languages.

3.1 Setup and evaluation

Data. For continual pre-training, we sample documents uniformly across high-resource languages (cf. §2.3) from the CulturaX dataset (Nguyen et al., 2024), resulting in a 35B-token training corpus. For our routing analysis, we use 500 held-out validation documents per language (both high- and low-resource) to compute the aggregated language-level expert usage distribution (cf. §2).

Training setup. We perform continual pre-training starting from OLMoE-Base on our 35B-token multilingual corpus and refer to the resulting

[3] Several of these languages are not traditionally considered low-resource (Joshi et al., 2020). We treat them as such to maintain a controlled experimental setup and ensure the availability of reliable evaluation benchmarks.

[4] See Appendix C for details on token overlap computation.

[5] We exclude English–Dutch as an adaptation pair: their token-vocabulary overlap is low (∼9%) and English routing is too diffuse to isolate language-specific experts.

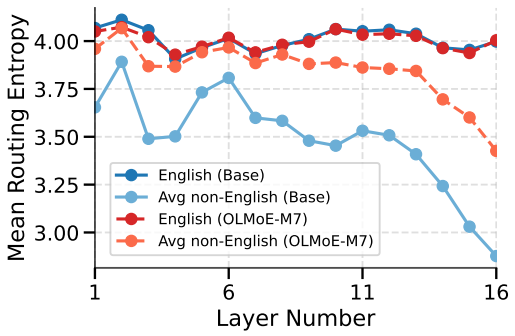


Figure 2: Comparison of routing entropy across layers for the English and non-English tokens for OLMoE-Base and OLMoE-M7.

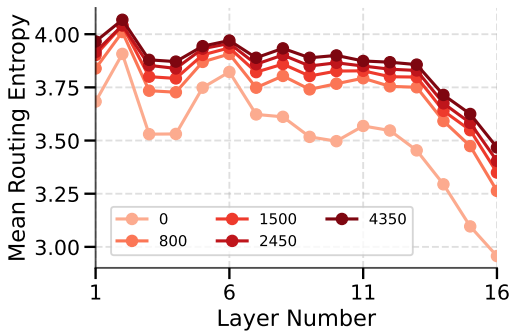


Figure 3: Routing entropy across layers at different steps of continual pre-training of OLMoE-M7 as indicated in the Legend. Lighter means earlier in step.

model as OLMoE-M7. Key architectural details and pre-training hyperparameters for this experiment are summarized in Appendix A.

Evaluation. We conduct both intrinsic and extrinsic evaluations to ensure the continual pre-training is successful. For intrinsic evaluation, we measure perplexity as a standard indicator of language modeling quality. For extrinsic evaluation, we evaluate all models on two multilingual downstream benchmarks: **Belebele** (Bandarkar et al., 2024) and **MultiBLiMP** (Jumelet et al., 2026). We select these datasets because they cover many languages from the CulturaX pre-training corpus. Moreover, both are widely used benchmarks to study multilingual model capabilities without requiring task-specific finetuning or instruction tuning, making them well suited for our setup (e.g., Foroutan et al., 2025; Messmer et al., 2026; Huang et al., 2026). More details can be found in Appendix F.

Results. Table 5 in Appendix B presents the intrinsic and extrinsic evaluations. Both perplexity and MultiBLiMP performance consistently and substantially improve with continual pre-training; only English degrades slightly, which is expected as the majority of tokens during continual pre-training are non-English. On Belebele, which is arguably a more difficult task, we observe much smaller, though still mostly consistent, improvements.

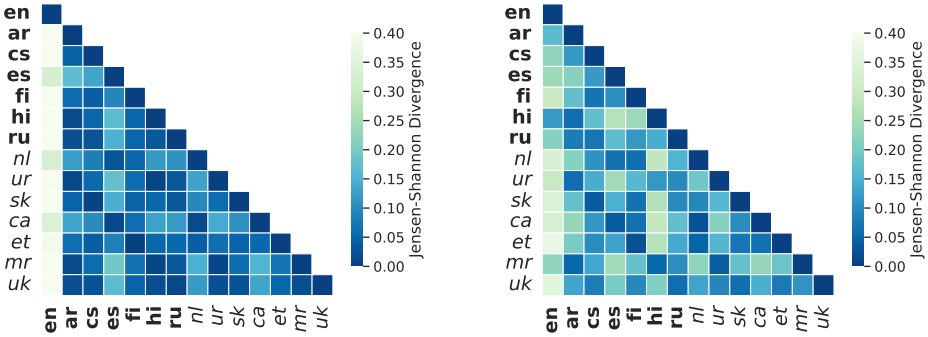
3.2 Qualitative findings for routing dynamics

We now analyze the effect of continual pre-training on the model's routing behavior using the metrics introduced in §2: entropy and JSD. The routing entropy signals how diffused expert activation is per language, but does not directly compare languages one-to-one. The JSD compares language pairs quantifying the extent to which their expert usage differs. We use these metrics to compare OLMoE-M7 to the OLMoE-Base model.

OLMoE has a dedicated routing pattern for English. To understand how routing evolves, we first analyze the starting state of the English-centric OLMoE-Base. Looking at entropy in Figure 2, we observe that the base model exhibits markedly lower entropy for non-English languages, particularly in the middle and later layers, which is consistent with findings from Bandarkar et al. (2026). This suggests that in the base model, a relatively small number of experts activates somewhat consistently for non-English languages, particularly in higher layers. This observation is corroborated by the pairwise JSD in the base model, presented in the Figure 4(a), which confirms that English tokens have very different routing patterns from all non-English languages in the final layer. In contrast, the divergence among non-English languages is consistently low, indicating that they are routed through a largely shared set of experts with little distinction between them. This pattern persists across other layers (see Figures 11 and 12 in Appendix D).

Continual pre-training diffuses expert usage across languages and language-specific routing occurs predominantly in final layers. When tracking entropy over continual pre-training steps (Figure 3), we find that the entropy for non-English languages gets closer to (but does not quite approach) the English-level entropy.[6] Additionally, Figure 5 shows how the JSD changes during training, per layer. The between-language divergence is lowest in the first decoder layer; it increases in a non-monotonic manner across the middle layers,

[6] A summary of entropy changes for individual high-resource languages is shown in Figure 10 in Appendix D.



(a) OLMoE-Base - Layer 15

(b) OLMoE-M7 Model - Layer 15

Figure 4: **Cross-lingual Routing Divergence in the Final Layer** using Pairwise JSD for OLMoE-Base (left) and OLMoE-M7 (right). Darker blue indicates higher expert sharing. **Bolded languages** are high-resource; *italicized languages* are low-resource.

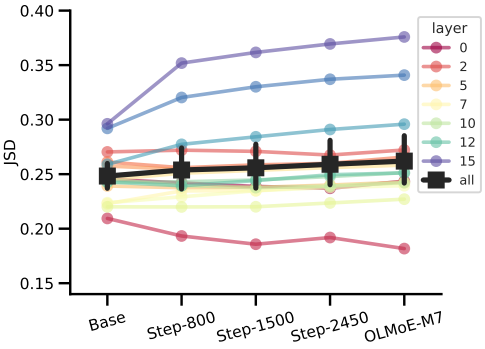


Figure 5: Mean JSD over training steps, showing increased language specialization in final decoder layers.

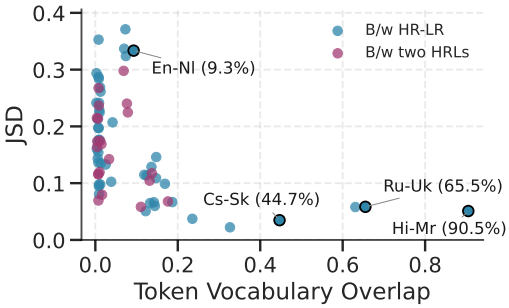


Figure 6: JSD vs Token-Vocabulary Overlap between language pairs. Each point is a language pair and the outlined points with black edges indicate a few qualitative examples for high and low-resource language pairs.

and becomes consistently higher towards the final layers. Over the course of training, JSD remains largely stable across the middle layers, decreases slightly in the earliest layer, and increases mainly in the final layers. Together, these statistics suggest that in the final layers, experts specialize more as evidenced by a larger change in JSD and lower entropy compared to other layers.

Based on these results, we conclude that, contrary to our hypothesis, multilingual continual pretraining does not necessarily induce *language separation* in the expert activation (except for the final layers), although it does *diffuse* them. Potentially, this is due to the fact that experts activate for multiple (related) languages at a time.

Vocabulary overlap drives differentiation more than language family. To better understand differentiation in the final layers, Figure 4(b) shows a heatmap of pairwise JSD in layer 15. Some language pairs exhibit highly similar routing patterns (e.g., Czech–Slovak, Finnish–Estonian), while others are strongly separated (e.g., Hindi–Catalan, or English, which differs from many languages). In these layers, routing similarity aligns closely with token-level vocabulary overlap, which can sometimes supersede typological markers in driving the router's statistical behavior (e.g., the high routing similarity between Hindi and Marathi vs. the divergence between English and Dutch). These pairs share roughly 90% vs. 9% of tokens in vocabulary, respectively, the latter despite a shared family. Figure 6 plots pairwise JSD against token-level vocabulary overlap for all language pairs. We find a moderate rank correlation (Spearman's $\rho = -0.56$), indicating that higher token overlap corresponds to more similar routing behavior.[7] Other measures of language similarity—syntactic, phonological, family-based, and geographic similarity from the `lang2vec` database (Littell et al., 2017; Malaviya et al., 2017)—all underperform in comparison (ρ of 0.30, 0.36, 0.11 and 0.46, respectively). Potentially, statistical vocabulary overlap supersedes the relevance of more traditional markers of language similarity in MoE routing. This could mean that prior MoE

[7] See Figure 9 in Appendix C for full statistics.

adaptation methods for low-resource languages may have been suboptimal (Zheng et al., 2025).

Finally, we note that Figure 4 demonstrates that English stands out in terms of routing behavior in the base *and* the continually trained model. The continual pre-training did reduce English-dominated routing patterns, yet only partially. To further investigate how vocabulary overlap affects routing, we examine how identical subword tokens shared by two languages are routed in the final layer. Routing agreement is high for related language pairs: 0.86 for Czech–Slovak, 0.87 for Russian–Ukrainian, 0.84 for Hindi–Marathi, and 0.82 for Spanish–Catalan. In comparison, Czech–Hindi, an unrelated control pair, has substantially lower agreement (0.64). These results suggest that vocabulary overlap affects cross-lingual routing partly through token identity: the same subword tends to be routed to similar experts across related languages.

In sum, these analyses indicate that multilingual continual pre-training gradually reshapes routing behavior. Rather than inducing clear language-specific expert separation throughout the model, training primarily diffuses expert usage across languages, with more pronounced differentiation in the final layers. In these layers, routing similarity is better explained by vocabulary overlap than by typological or family-based language similarity. We observe the same qualitative routing trends in the Phi-3.5-MoE model; see Appendix E.

4 Low-Resource Adaptation

Motivated by our findings in §3, which indicate consistent co-routing of high-overlap language pairs in the final decoder layers, we now investigate whether the routing dynamics of OLMoE-M7 can be leveraged to adapt the model to related low-resource languages in a parameter-efficient manner.

Hypothesis: MoE models that are specialized for a high-resource language in the final decoder layers can be leveraged to efficiently improve performance on a related low-resource language.

4.1 Adaptation methods

We first introduce our adaptation methods.

Routing-aware expert selection. Testing our hypothesis requires an expert selection procedure, which, given a target language, returns a subset of experts to train. We propose a method based on what we term *activation gap* (see Figure 7 for a schematic illustration). For each expert and language the model has been continually trained on, we first compute normalized activation frequencies: the fraction of tokens of that language for which the expert is selected among the top-k routed experts at its layer, normalized by the total number of tokens of that language. Next, we identify the two most activating languages for each expert and compute the activation gap as the difference between their activation frequencies. We consider top-2 languages because the second-highest captures the strongest competing language for an expert.

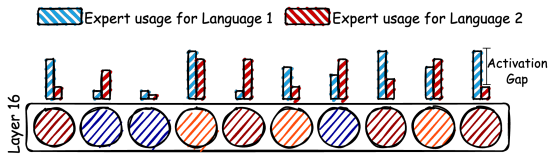


Figure 7: Illustration of the "activation gap" procedure. Specialized experts are identified via the difference in activation frequency between their top two languages.

Selective expert finetuning. We introduce *Selective Expert Finetuning* (SEFT), which adapts only a small subset of experts that are most strongly associated with a target low-resource language. SEFT adopts the activation gap approach described above to decide which experts are most relevant for the target language. Specifically, we select all experts in the final two layers, where our analysis in §3 shows the strongest language-specific routing differentiation, as evidenced by the highest cross-lingual JSD (Figure 5)—that exceed an activation gap threshold of 1% (see Table 6 in Appendix G for a sensitivity analysis on this threshold). This typically results in selecting 5 to 7 experts per layer.

Selective and shared expert finetuning. For this method, we augment the language-dominant experts selected by SEFT with a small set of experts that are highly active across all languages and term this approach *Selective and Shared Expert Finetuning* (SSFT). Specifically, for each of the final two layers we add five such shared experts for each layer to the finetuning pool and update them jointly with the language-specific experts. We ablate the number of shared experts k in Appendix G, Table 7.

Baselines. To validate that gains from SEFT and SSFT arise from meaningful expert specialization rather than parameter count alone, we introduce

a *Random Expert Finetuning* (Random-SEFT) baseline, where a randomly selected set of experts matched in number to SEFT is finetuned. To ensure that improvements are not simply due to finetuning a larger number of experts, we conduct an additional control experiment called SEFT-Top20. Here, we expand the SEFT expert set using the same activation-gap ranking to include approximately 30% of experts (top-20) in each of the final two layers. This allows us to disentangle the effects of expert quantity from expert specificity. As another baseline, we also consider *All-Experts Finetuning* (AEFT), where all experts and router parameters in the final two layers are updated, testing whether broader expert adaptation provides additional benefits beyond selective specialization. As an upper bound, we include *Full-Model Finetuning* (Full-FT), in which all model parameters are updated.

Crucially, across all parameter-efficient strategies, only the selected experts and routers for respective layers are trainable. All other experts, attention layers, and embeddings remain frozen. These methods update approximately 75–251M parameters ($\sim$1.1–3.6% of the 7B total parameters, or $\sim$7.5–25.1% relative to the $\sim$1B active parameters). In contrast, AEFT updates 800M parameters ($\sim$11.4% of total parameters), while Full-FT updates all 7B parameters.

LoRA baseline. We additionally compare SSFT against LoRA applied to the last two layers, using rank $r = 128$, which results in approximately 151M trainable parameters, slightly more than SSFT's 138M. We use the same learning-rate sweep and model-selection protocol as for the other adaptation methods.

4.2 Experimental Setup

Next, we detail the experimental setup for our low-resource adaptation experiments.

Data. For each low-resource target we assume access to a related high-resource "anchor" language seen during continual pre-training, and use the anchor's routing behavior to identify which experts to adapt. We adopt the six low-resource targets and their high-resource anchors introduced in §2.3. To simulate a low-resource setting, we sample approx. 300M tokens per language, corresponding to roughly 5% of the multilingual pretraining budget.

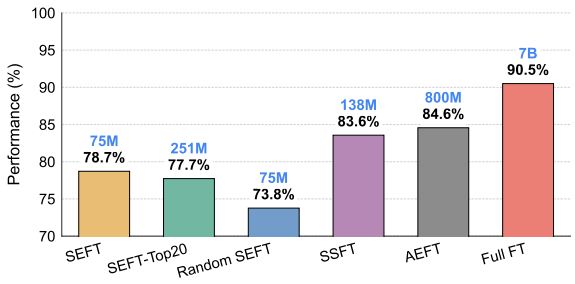


Figure 8: Average **MultiBLiMP** performance across target languages comparing different adaptation strategies (SEFT, SSFT) against baselines. Numbers in blue next to each bar indicate trainable parameters.

Hyperparameters. We sweep the learning rate over the set $\{1e{-}5, 1e{-}4, 4e{-}4, 1e{-}3, 4e{-}3\}$ for all methods. Model selection is based on perplexity measured on a held-out validation set for each low-resource language. For all methods, we report results for the checkpoint achieving the lowest validation perplexity.

4.3 Results

We report results on the two multilingual benchmarks introduced in §3, MultiBLiMP and Belebele (cf. Appendix F for details). At our model scale and training setting, results on Belebele are less consistent than on MultiBLiMP, with scores closer to the random baseline; we therefore treat MultiBLiMP, together with perplexity, as the primary signal for comparing adaptation strategies.

Finetuning language-specific experts outperforms finetuning random experts. We first validate the effectiveness of our routing-aware expert selection finetuning by comparing SEFT to Random-SEFT, which adapt the same number of parameters. On MultiBLiMP, Figure 8 shows that Random-SEFT yields the lowest average performance across all target languages (73.8% $\pm$ 0.49 over three random expert selections), substantially lagging behind SEFT (78.7%) and other routing-aware strategies.

Shared experts improve overall performance. Targeting language-specific experts provides gains over Random-SEFT, and the remaining results in Figure 8 furthermore show that incorporating shared experts further improves performance: SSFT yields an average performance of 83.6%, outperforming SEFT by 4.9 percentage points, with average performance generally improving as the number of shared experts k increases (Table 7 Ap-

pendix G). Looking at individual languages, in Table 1, SSFT leads to substantial performance improvements for, e.g., Marathi (+7.2 points) and Estonian (+6.9). To test whether performance gains are driven simply by the increased numbers of trainable parameters or by the nature of the experts, we compare SSFT against SEFT-top20, a baseline where we increase the number of language-specific experts updated without including shared experts. Despite updating a larger budget of experts, SEFT-Top20 achieves only 77.7% average accuracy, falling well short of SSFT (83.6%). On Belebele (Table 2), SSFT again outperforms SEFT on average, with the largest gains for Estonian (+4.2 points) and Catalan (+1.9). To better understand what these shared experts capture, we inspect their top-activating tokens in the final layers. We find that they primarily process language-agnostic subword units, such as individual letters, short character sequences, and word fragments shared across languages. In contrast, punctuation and function words account for approximately 32% of tokens routed to language-specific experts, compared with approximately 13% for shared experts.

Comparison with LoRA. We further compare SSFT against LoRA at a comparable parameter budget. LoRA with rank $r = 128$ updates approximately 151M parameters, compared with 138M for SSFT. Averaged across target languages, SSFT outperforms LoRA by 0.7 points on MultiBLiMP and 0.8 points on Belebele, while updating fewer parameters and without introducing additional parameters. LoRA could also be combined with SSFT, which we leave to future work.

Computational advantage. For OLMoE-1B, full Full-FT on 300M tokens requires approximately 2 hours on 16 H100 GPUs ($\approx$ 32 GPU-hours), corresponding to $\approx 9.8 \times 10^{17}$ FLOPs. In contrast the parameter-efficient adaptation methods, SSFT and SEFT update only 75-138M parameters and complete training in 45-50 minutes on 4 H100s ($\approx$ 3.3 GPU hours), utilizing $\approx 9.8 \times 10^{15}$ FLOPs. This leads to a 10x reduction in GPU-hours and 100x reduction in FLOPs highlighting SSFT's computational efficiency in comparison to Full-FT.[8]

[8]While Full-FT provides the overall strongest performance when adapting on 300M tokens, we note that it leads to catastrophic forgetting for the languages adapted to in the initial continual pre-training stage when training on > 800M tokens in the subsequent stage of adaptation. SEFT on the other hand, leads to less catastrophic forgetting. We provide further details on these results in Appendix H.

Lang.	SEFT	SSFT	LoRA	AEFT	Full FT	M7
ca	82.9	85.4	87.0	89.0	93.3	81.3
et	68.4	75.3	71.0	74.0	81.9	61.5
mr	66.5	73.7	75.2	74.6	81.7	68.5
sk	86.3	88.6	89.3	89.9	92.7	86.0
uk	81.7	85.1	82.7	85.5	91.9	80.7
ur	86.5	93.3	92.4	94.4	96.5	83.3

Table 1: Results on **MultiBLiMP** for target languages using different adaptation strategies.

Lang.	SEFT	SSFT	LoRA	AEFT	Full FT	M7
ca	29.2	31.1	30.1	29.2	34.9	31.4
et	26.0	30.2	26.9	29.6	30.6	25.3
mr	29.9	27.7	27.2	26.2	30.9	25.8
sk	33.8	34.3	34.0	33.7	34.6	33.0
uk	31.6	31.3	30.8	32.2	35.0	30.3
ur	28.3	28.9	29.6	29.2	27.8	28.9

Table 2: **Belebele** (4-shot) performance for target languages using different adaptation strategies.

5 Related Work

Multilingual MoE Analysis. Prior work has studied expert specialization in multilingual MoE models, though evidence for language-specific modularity remains mixed and is largely limited to encoder-based or sequence-to-sequence architectures. Zoph et al. (2022) find that encoder MoEs tend to specialize over shallow token groupings rather than by language, casting doubt on whether multilingual MoEs can reliably form language-specific experts. In multilingual neural MT, Kudugunta et al. (2021) propose routing strategies at multiple granularities, but do not analyze expert specialization across languages. Similarly, Meta's NLLB MoE models demonstrate the scalability of MoEs for translation, yet do not investigate how experts specialize or how routing patterns vary across languages (Team et al., 2022). Zheng et al. (2025) leveraged MoE modularity primarily for medical domain specialization, introducing language-family experts and hybrid routing strategies that utilize late-stage language specialization but rely on explicit language identity at inference time. Bandarkar et al. (2026) study routing dynamics in MoE-based LLMs, showing that experts are language-specific in early and late layers but largely shared in middle layers. They analyze pre-trained models only and propose a steering method which is applied during inference to improve multilingual generalization. In contrast, we focus on training-based adaptation. Complementary to these analyses, Kallini et al. (2025) study how subword

vocabulary overlap shapes cross-lingual representation alignment and transfer in bilingual models; we instead probe the mechanical routing decisions that overlap induces in multilingual MoEs.

Efficient Multilingual Adaptation. Previous work aims to mitigate the 'curse of multilinguality' through modularity and resource optimization. Frameworks like MAD-X (Pfeiffer et al., 2020) and MAFT (Alabi et al., 2022) utilize adapters and continual pre-training to adapt models to new languages, while Ansell et al. (2022) perform adaptation via parameter-efficient sparse masks. Marchisio et al. (2023) propose 'mini-model' training to reduce compute costs, and Gurgurov et al. (2024) integrate external knowledge graphs to compensate for low resource languages.

6 Conclusion

We studied the routing dynamics of MoE models in multilingual continual pre-training. Our analysis reveals that multilingual adaptation leads to diffuse, language-agnostic routing throughout the early and middle layers of the network. Distinct language specialization emerges gradually in the final layers, where co-routing of languages correlates more strongly with token-level vocabulary overlap than with language families. While our experiments focus on the OLMoE architecture, the routing dynamics and specialization trends across layers align with findings from MoE and dense model literature (Bandarkar et al., 2026; Kojima et al., 2024).

Leveraging these insights, we proposed Selective and Shared Expert Finetuning (SSFT), a parameter-efficient adaptation strategy. By updating only the language-dominant and shared experts in the final layers, SSFT achieves strong performance on benchmarks like MultiBLiMP and Belebele, while updating less than 2% of the model parameters. Hence SSFT offers computational advantages compared to full fine-tuning and can be more robust to catastrophic forgetting. Overall, our findings suggest that effective low-resource adaptation of MoE models relies on both targeting specialized experts for new languages and preserving shared experts to maintain cross-lingual stability. While this work focuses on low-resource adaptation, the observed dynamics of specialization and cross-lingual expert sharing are broadly applicable to the study of multilingual MoEs and could naturally inform more general cross-lingual transfer strategies or efficient modular model adaptation.

Limitations

Due to the substantial computational demands of continual pre-training, we conduct our controlled training experiments on OLMoE-Base (1B active / 7B total parameters). While we reproduce the main qualitative routing trends on Phi-3.5-MoE model (Appendix E), validation across additional model scales and architectures remains future work. Our proposed adaptation strategy (SSFT) relies on the existence of a high-resource "anchor" language with significant vocabulary overlap to identify relevant experts. This approach may prove less effective for language isolates or low-resource languages that lack a close high-resource relative in the pre-training data. We examine routing dynamics specifically within the context of continual pre-training on a balanced multilingual corpus (35B tokens). The emergence of language specialization in the final layers may differ under alternative training regimens, such as curriculum learning, different data mixing ratios, or during pre-training from scratch.

Acknowledgements

We thank the members of our McGill and Mila research labs for their feedback throughout the course of this project. We especially thank Jay Gala and Harman Singh for their thoughtful reviews and valuable discussions that helped improve this work. Finally, we thank the Compute Canada and Mila IT support teams for their continuous assistance and for providing the computational resources necessary to run our experiments.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024a. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024b. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1

others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Jesujoba Alabi, Marius Mosbach, Matan Eyal, Dietrich Klakow, and Mor Geva. 2024. The hidden space of transformer language adapters. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6607.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349.

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.

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.

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. 2026. Multilingual routing in mixture-of-experts. In *The Fourteenth International Conference on Learning Representations*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. DeepSeek-V3 Technical Report. *Preprint*, arXiv:2412.19437.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Negar Foroutan, Paul Teiletche, Ayush Kumar Tarun, and Antoine Bosselut. 2025. Revisiting multilingual data mixtures in language model pretraining. *Preprint*, arXiv:2510.25947.

Daniil Gurgurov, Mareike Hartmann, and Simon Ostermann. 2024. Adapting multilingual LLMs to low-resource languages with knowledge graphs via adapters. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pages 63–74.

Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, and 1 others. 2026. A survey on large language models with multilingualism: Recent advances and new frontiers. *Artificial Intelligence Review*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of Experts. *Preprint*, arXiv:2401.04088.

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.

Jaap Jumelet, Leonie Weissweiler, Joakim Nivre, and Arianna Bisazza. 2026. Multiblimp 1.0: A massively multilingual benchmark of linguistic minimal pairs. *Transactions of the Association for Computational Linguistics*, 14:193–216.

Julie Kallini, Dan Jurafsky, Christopher Potts, and Martijn Bartelds. 2025. False Friends Are Not Foes: Investigating Vocabulary Overlap in Multilingual Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21138–21154.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6919–6971.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Thang Luong, and Orhan Firat. 2021. Exploring routing strategies for multilingual mixture-of-experts models.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, and 40 others. 2024. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 14200–14282.

Junzhuo Li, Bo Wang, Xiuze Zhou, Peijie Jiang, Jia Liu, and Xuming Hu. 2025. Decoding knowledge attribution in mixture-of-experts: A framework of basic-refinement collaboration and efficiency analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22431–22446.

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

Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 8–14.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2025. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4427–4447.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535.

Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. 2023. Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5474–5490.

Bettina Messmer, Vinko Sabolčec, and Martin Jaggi. 2026. Enhancing multilingual LLM pretraining with model-based data selection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, and 1 others. OLMoE: Open mixture-of-experts language models. In *The Thirteenth International Conference on Learning Representations*.

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

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.

Noam Shazeer. 2020. GLU variants improve transformer. *Preprint*, arXiv:2002.05202.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi K2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No Language Left Behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do Llamas work in English? On the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You.

2024. OpenMoE: An early effort on open mixture-of-experts language models. In *International Conference on Machine Learning*, pages 55625–55655. PMLR.

Guorui Zheng, Xidong Wang, Juhao Liang, Nuo Chen, Yuping Zheng, and Benyou Wang. 2025. Efficiently democratizing medical LLMs for 50 languages via a mixture of language family experts. In *The Thirteenth International Conference on Learning Representations*.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.

A Pre-training Hyperparameters

Table 3 provides the detailed hyperparameters and architectural specifications used for the continual pre-training of the OLMoE-Base model.

Hyperparameter	Value
Model architecture	Decoder-only MoE
Number of layers	16
Hidden dimension (d_{model})	2048
Attention heads	16
Activation function	SwiGLU (Shazeer, 2020)
Sequence length	4096
Vocabulary size	50,280
Number of experts per layer	64
Top-k routing	8
MoE routing type	Dropless, sparse MLP
Router z-loss weight	0.001 (Zoph et al., 2022)
Auxiliary MoE loss weight	0.01 (Shazeer et al., 2017)
Optimizer	AdamW
Learning rate	1×10^{-4}
Adam β coefficients	(0.9, 0.95)
Weight decay	0.1
Global batch size	2048
Precision	bfloat16 (AMP)

Table 3: Key hyperparameters used for continually pre-training OLMoE-Base.

B Continual Pre-training Results

Table 5 reports per-language intrinsic (perplexity) and extrinsic (MultiBLiMP, Belebele 4-shot) evaluations for OLMoE-Base and OLMoE-M7 across the seven high-resource pre-training languages. Continual multilingual pre-training substantially reduces perplexity and improves MultiBLiMP performance on all non-English languages, with a modest degradation on English; Belebele improvements are smaller but mostly consistent.

C Token-Vocabulary Overlap

We compute token-level vocabulary overlap by tokenizing parallel Bible chapters using the OLMoE tokenizer, leveraging the wide multilingual coverage of the Bible. Pairwise vocabulary overlap statistics are shown in Figure 9.

To verify that these overlap statistics are not specific to the Bible domain, we additionally compute token-level vocabulary overlap on the continual pre-training corpus. The two estimates are strongly correlated across language pairs (Pearson $r = 0.91$), with a mean absolute difference of approximately 6 percentage points. This suggests that the overlap–routing relationship is not an artifact of the Bible corpus.

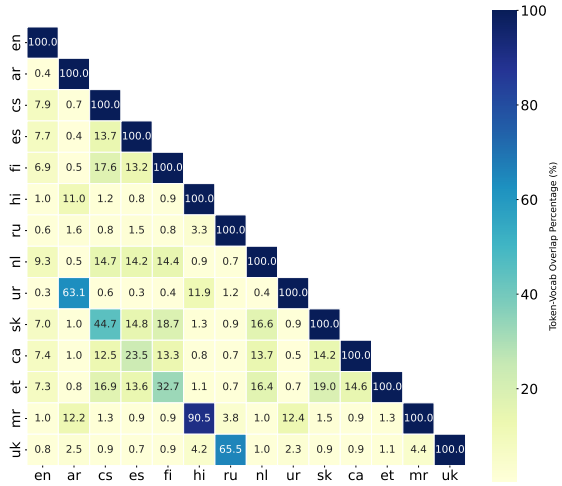


Figure 9: Token-Vocabulary Overlap across language pairs.

D Routing Analysis

We provide additional routing analysis specifically for the OLMoE-Base and OLMoE-M7 models. Figures 11 and 12 illustrate the pairwise Jensen-Shannon Divergence (JSD) for the base model and OLMoE-M7 in layers 13 and 14, supplementing the analysis in Section 3. Additionally, Figure 10 illustrates the routing entropy across different layers for OLMoE-Base and OLMoE-M7 models across high-resource languages.

E Routing Analysis on Phi-3.5-MoE

To examine whether our findings generalize beyond OLMoE, we repeat the routing analysis on Phi-3.5-MoE (Abdin et al., 2024a), a substantially larger model with 6.6B active and 42B total parameters, 16 experts with top-2 routing, and 32 layers. We observe the same qualitative pattern: routing remains diffuse through the early and middle layers, with entropy around 2.2–2.4 nats, before becoming substantially more concentrated in the final layer, where entropy falls to 1.45 nats. Final-layer routing divergence also correlates with token-vocabulary overlap (Spearman $\rho = -0.67$, $p < 10^{-12}$), compared with $\rho = -0.56$ for OLMoE-M7. These results suggest that the observed late-layer specialization and overlap-driven co-routing are not specific to the OLMoE architecture.

F Evaluation Benchmarks

We evaluate our adaptation strategies on Belebele and MultiBliMP: Belebele is a multiple-choice machine reading comprehension benchmark covering

Language	English	Arabic	Czech	Spanish	Finnish	Hindi	Russian	Target	Original Target
OLMoE-M7	97.7	96.0	94.0	97.1	96.3	99.1	96.0	-	-
SSFT Results									
Catalan	98.2	95.3	93.9	95.1	95.6	99.2	95.8	85.9	81.3
Estonian	98.1	94.9	93.1	95.6	94.5	99.1	95.2	77.0	61.5
Marathi	96.4	**80.7**	**74.4**	**88.0**	**81.4**	**97.2**	**71.1**	79.1	68.5
Slovak	97.9	95.1	93.0	96.3	96.0	99.2	95.6	89.0	86.0
Ukrainian	97.9	95.1	92.6	97.0	95.8	99.2	95.0	83.5	80.7
Urdu	97.9	94.2	93.3	97.2	96.3	99.0	95.6	92.4	83.3
Full FT Results									
Catalan	96.9	94.8	**91.6**	**94.3**	**93.1**	99.4	96.2	95.8	81.3
Estonian	97.7	95.1	**90.9**	**94.6**	**86.1**	98.9	95.7	90.9	61.5
Marathi	97.4	94.0	93.3	96.6	**94.6**	**91.7**	95.7	89.1	68.5
Slovak	97.5	94.2	**90.5**	**96.0**	93.7	99.0	96.4	93.8	86.0
Ukrainian	97.4	95.1	92.8	96.8	95.4	99.3	92.1	93.0	80.7
Urdu	97.7	**89.5**	94.3	96.8	96.2	99.2	95.6	96.7	83.3

Table 4: MultiBliMP performance on High-Resource languages used to train OLMoE-M7 and effect of SSFT and Full Finetuning by training on 800M tokens. Highlighted cells denote cases where performance drops by more than one standard deviation of the downstream performance relative to OLMoE-M7 continually trained on high-resource languages.

Language	Perplexity ↓		MultiBLiMP ↑		BeleBele (4-shot) ↑	
	OLMoE-Base	OLMoE-M7	OLMoE-Base	OLMoE-M7	OLMoE-Base	OLMoE-M7
English (eng)	**12.48**	13.36	**0.981**	0.977	**0.590**	0.390
Arabic (arb)	5.68	**3.00**	0.828	**0.960**	0.307	**0.344**
Czech (ces)	11.33	**4.45**	0.758	**0.940**	**0.354**	**0.354**
Spanish (spa)	10.18	**6.78**	0.928	**0.971**	0.309	**0.364**
Finnish (fin)	17.82	**5.01**	0.753	**0.963**	0.316	**0.328**
Hindi (hin)	3.20	**1.97**	0.930	**0.991**	0.296	**0.328**
Russian (rus)	7.04	**4.22**	0.870	**0.960**	0.354	**0.368**

Table 5: Performance on high-resource languages before and after multilingual continual pretraining. OLMoE-M7 improves performance across all non-English languages, with a modest degradation for English.

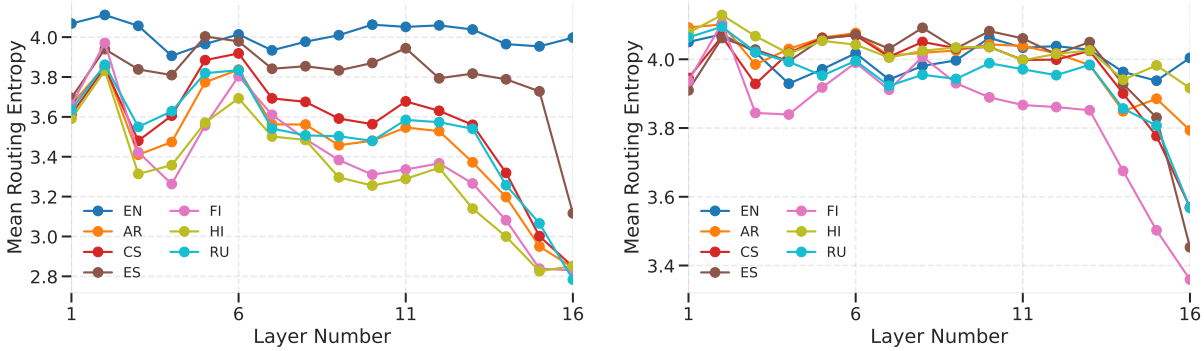
122 language variants, with questions grounded in short passages from FLORES-200 (Team et al., 2022), enabling direct cross-lingual comparison of semantic understanding. MultiBLiMP is a multilingual benchmark of syntactic and morphological minimal pairs generated using Universal Dependencies and UniMorph, evaluating sensitivity to fine-grained grammatical distinctions across languages.

G Sensitivity to Expert Selection Hyperparameters

We provide two ablations supporting the hyperparameter choices in SEFT and SSFT: the activation gap threshold α (Table 6) and the number of shared experts k (Table 7).

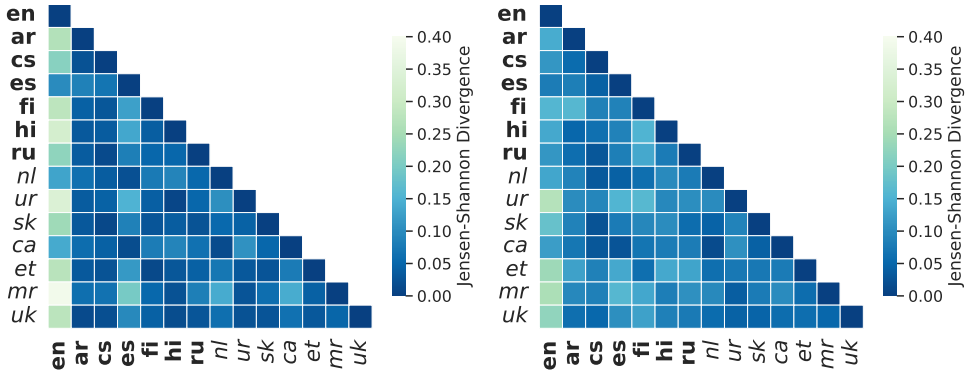
Activation gap threshold (α). The 1% activation gap acts as a filter: if it is too low, we admit "noisy" experts that lack language-specific dominance (similar to our SEFT-Top20 control); if it is too high, the selected expert pool shrinks dramatically and few experts remain per language, yielding little change relative to the baseline. Table 6 reports SEFT performance on MultiBLiMP when varying $\alpha \in \{2\%, 1\%, 0.0001\%\}$. The strict 2% setting leaves the model with too few experts, most evidently for Marathi, where no experts meet the threshold, while the relaxed 0.0001% setting fails to construct a meaningful expert pool, yielding inconsistent gains. The $\alpha = 1\%$ setting used in the main paper provides the best balance.

Number of shared experts (k). The shared experts in SSFT are identified by computing the mean activation of every expert across all seven high-resource pre-training languages, using a held-out validation set of 5,000 samples per language; we then select the k experts with the highest



(a) OLMoE-Base (b) OLMoE-M7

Figure 10: Comparison of routing entropy across layers for OLMoE-Base (left) and OLMoE-M7 (right) across all high-resource languages.



(a) OLMoE-Base - Layer 13 (b) OLMoE-M7 - Layer 13

Figure 11: **Cross-lingual Routing Divergence in Layer 13** using Pairwise Jensen–Shannon Divergence (JSD)for OLMoE-Base (left) and OLMoE-M7 (right). Darker blue indicates higher expert sharing.

Lang.	α=2%	α=1%	α=0.0001%	M7
Cat.	81.9	82.9	83.5	81.3
Est.	65.0	68.4	72.8	61.5
Mar.	n/a	66.5	67.4	68.5
Slk.	85.7	86.3	86.8	86.0
Ukr.	79.6	81.7	80.4	80.7
Urd.	82.2	86.5	82.2	83.3

Table 6: MultiBLiMP performance for SEFT under different activation gap thresholds α. "M7" is the OLMoE-M7 baseline before adaptation. "n/a" indicates that no experts met the threshold (Marathi at α=2%).

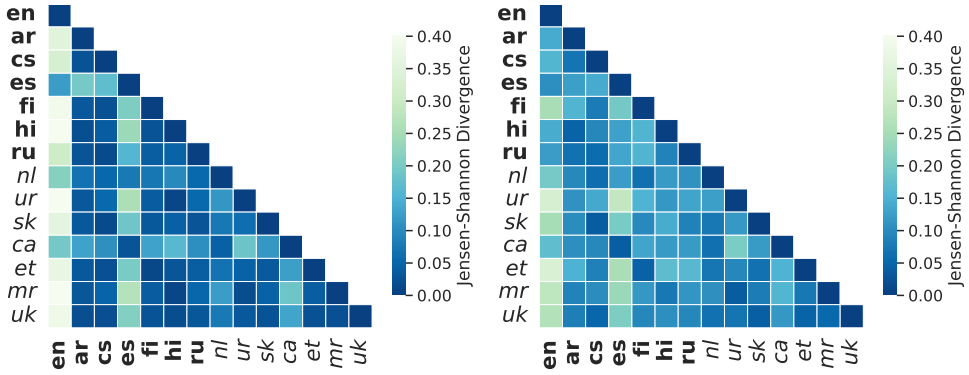
Lang.	k=0	k=1	k=3	k=5	M7
Cat.	82.9	83.2	84.2	85.4	81.3
Est.	68.4	71.5	72.9	75.3	61.5
Mar.	66.5	67.0	72.4	73.7	68.5
Slk.	86.3	86.5	88.0	88.6	86.0
Ukr.	81.7	79.3	78.1	85.1	80.7
Urd.	86.5	84.7	86.4	93.3	83.3

Table 7: MultiBLiMP performance for SSFT under varying numbers of shared experts k (k=0 corresponds to SEFT). "M7" is the OLMoE-M7 baseline before adaptation.

mean activation. To assess sensitivity, we vary $k \in \{0, 1, 3, 5\}$ (with k=0 recovering SEFT). Table 7 shows that average performance generally improves as k increases, although this trend is not monotonic for every individual language, suggesting that shared experts contribute beyond what language-specific experts provide alone. Combined with the SEFT-Top20 control in Figure 8, this indicates that the *sharedness* of these experts, not the added parameter count, drives the improvement.

H Catastrophic Forgetting

Having established SSFT as an effective and parameter-efficient adaptation strategy, we next examine whether it also improves training stability over full-finetuning. Using an extended 800M-token adaptation setting to amplify forgetting effects, we observe that full-model finetuning substantially degrades performance on MultiBlimp previously seen languages during continual pretrain-



(a) OLMoE-Base - Layer 14

(b) OLMoE-M7 - Layer 14

Figure 12: **Cross-lingual Routing Divergence in Layer 14** using Pairwise Jensen–Shannon Divergence (JSD) for OLMoE-Base (left) and OLMoE-M7 (right). Darker blue indicates higher expert sharing.

ing, whereas SSFT largely preserves performance on those languages as shown in Table 4. This suggests that selective experts finetuning acts as a form of regularization enabling adaptation without overwriting already-learnt representations.

I Licenses of Scientific Artifacts

We list the licenses of the scientific artifacts used in this work. The CulturaX dataset is released under the CC0-1.0 and ODC-BY licenses. The OLMoE model is released under the Apache 2.0 license. Our use of these artifacts is consistent with their intended use and licensing terms.