Unveiling Language Routing Isolation in Multilingual MoE Models for Interpretable Subnetwork Adaptation

Anonymous ACL submission

Abstract

Mixture-of-Experts (MoE) models exhibit striking performance disparities across languages, yet the internal mechanisms driving these gaps remain poorly understood. In this work, we conduct a systematic analysis of expert routing patterns in MoE models, revealing a phenomenon we term *Language Routing Isolation*, in which high- and low-resource languages tend to activate largely disjoint expert sets. Through layer-stratified analysis, we further show that routing patterns first converge and then diverge across model depth. Building on these findings, we propose **RISE** (routing isolation-guided subnetwork enhancement), a framework that identifies and adapts language-specific expert subnetworks. RISE uses a tripartite selection strategy: specificity scores select language-specific experts in shallow and deep layers, while overlap scores select universal experts in middle layers. By training only the selected subnetwork while freezing the rest of the model, RISE improves low-resource performance while preserving other-language capabilities. Across 10 languages, RISE achieves target-language F1 gains of up to 10.85% with minimal cross-lingual degradation. Our anonymized code repository is available at: `https://anonymous.4open.science/r/rise-5C2C/`.

1 Introduction

The MoE paradigm has emerged as a powerful architectural approach in modern language modeling, demonstrating remarkable capabilities across a wide range of tasks, including reasoning, code generation, and multilingual understanding (Abdin et al., 2024; DeepSeek-AI et al., 2024b,a). By leveraging sparse expert activation, MoE models enable scaling to tens or hundreds of billions of parameters while maintaining practical inference efficiency, making them particularly attractive for resource-intensive applications. Despite their empirical success, MoE models exhibit significant performance disparities across languages (Chua et al., 2025). As shown in Table 1, MoE models show accuracy gaps exceeding 80 percentage points on the MGSM benchmark between high-resource languages (e.g., English) and low-resource languages (e.g., Bengali and Swahili).

Recent studies have begun to analyze multilingual MoE routing from different statistical views. Bandarkar et al. (2026) characterize layer-wise routing divergence by *averaging post-softmax routing weights* across tokens in parallel sequences, showing that non-English routing often diverges from English in early and late layers while becoming more aligned in middle layers. Complementarily, Chen et al. (2026) study expert-selection frequencies and language-related experts, revealing family-structured routing shared/exclusive expert patterns. These works provide important diagnostic evidence that multilingual routing is structured, but leave open an adaptation-oriented question: which experts are actually traversed by a target language, how isolated are these executable expert supports across languages, and how should such global and layer-wise structure guide parameter updates?

To address this question, we analyze multilingual routing behavior from a discrete execution-support perspective, counting the experts selected by the Top-K router for each token. This view directly corresponds to the sparse computation path of the MoE model, naturally supports set-based intersection/union analysis across languages, and is

less affected by small changes in non-selected or low-ranked routing weights. Using this execution-support view, we study multilingual routing at two complementary granularities. At the global level, we characterize the *routing isolation* phenomenon, where high- and low-resource languages tend to occupy weakly overlapping expert-support regions. Through layer-stratified analysis, we further reveal *layer-wise convergence–divergence*: routing converges in middle layers, which are more language-agnostic, but diverges in shallow and deep layers, which capture language-specific encoding and generation. Together, the global and layer-wise views clarify how executable expert supports are organized across languages and provide a principled basis for multilingual adaptation in practice.

Building on these findings, we propose RISE (**R**outing **I**solation-guided **S**ubnetwork **E**nhancement), a method designed to improve low-resource language performance while preserving the model's capabilities in high-resource languages.

RISE adopts a hierarchical selection strategy to identify the expert subnetwork that predominantly supports the target language. Specifically, in shallow and deep layers, RISE uses a specificity score to identify experts that are particularly associated with the target low-resource language. In middle layers, where routing patterns are more shared across languages, RISE uses an overlap score to select universal experts. Extensive experiments show that RISE consistently improves target-language performance while preserving performance on other languages and tasks.

In summary, this paper makes the following contributions:

❶ **Empirical finding.** We introduce a discrete execution-support analysis of multilingual MoE routing, jointly examining global expert-set overlap and layer-wise routing dynamics. This analysis characterizes two routing patterns, *routing isolation* and *layer-wise convergence–divergence*, that expose how executable expert supports are organized across languages and motivate our adaptation strategy in RISE.

❷ **Methodological contribution.** Building on these observations, we propose RISE, a layer-stratified expert selection method for low-resource language adaptation that selects distinctive experts in shallow and deep layers while preserving shared experts in the middle layers.

❸ **Empirical validation.** We conduct comprehensive experiments across 10 languages using multiple datasets, validating both the effectiveness and efficiency of our routing-isolation approach.

2 Related Work

MoE routing and expert specialization. Understanding how sparse experts specialize has been a central question in MoE interpretability and adaptation. Prior work has analyzed expert utilization, routing concentration, and expert contribution in large MoE models (Shazeer et al., 2017; Chaudhari et al., 2026). Recent studies further show that stronger expert specialization can be encouraged through routing- or objective-level design, such as expert orthogonality, routing diversification, and expert-specialized fine-tuning (Guo et al., 2026; Wang et al., 2024). These works motivate the view that expert routing is not only a computational mechanism, but also a useful handle for understanding and adapting model behavior.

Multilingual and language-guided MoE routing. Multilingual MoE models expose an additional structure: routing can vary systematically across languages. Bandarkar et al. (2026) analyze multilingual routing in MoE LLMs and report that routing is often more language-specific in shallow and deep layers, while middle layers show stronger cross-lingual alignment. In multilingual machine translation, Zhao et al. (2024) introduce language-guided routing that incorporates linguistic hierarchy and adaptively allocates experts across languages. Zheng et al. (2025) propose language-family experts for multilingual medical LLMs, showing that linguistic priors can guide scalable expert sharing across many languages. These studies indicate that language structure is deeply entangled with expert routing.

Our positioning. Notably, our work is close to multilingual routing analyses such as Ban-

Model	BN	DE	EN	ES	FR	JA	RU	SW	TH	ZH	Avg.
Phi-3.5-MoE-instruct	1	79	88.5	81.5	73	56	77	1	26.5	65.5	54.9
Qwen3-30B-A3B	46	88.5	96.5	91.5	82	83.5	92.5	48	87.5	86	80.3

Table 1: Accuracy on MGSM across languages. Language codes: BN (Bengali), DE (German), EN (English), ES (Spanish), FR (French), JA (Japanese), RU (Russian), SW (Swahili), TH (Thai), ZH (Chinese).

darkar et al. (2026), but differs in goal. Rather than treating routing statistics only as a diagnostic object, we use them as an operational signal for selective adaptation: RISE identifies a target-language expert subnetwork in an existing pretrained MoE LLM and updates only that subnetwork. Compared with language-guided MoE training or architecture design (Zhao et al., 2024; Zheng et al., 2025), RISE does not modify the router or add language-family experts; compared with general expert-specialized fine-tuning (Wang et al., 2024), it uses layer-aware multilingual routing structure to choose and decide which experts to adapt.

3 Routing Analysis

To understand the multilingual capability disparities in MoE models, we analyze expert routing patterns from two complementary perspectives: global-level activation statistics and layer-wise routing dynamics. These two views operate at different granularities and are complementary rather than contradictory: routing isolation is defined over the globally aggregated top-K routed support across all layers, whereas the layer-wise analysis decomposes similarity at each depth relative to a high-resource reference language. We use the same data from the multi-lingual benchmarks for analysis and experiments in the next section.

3.1 *Routing Isolation* at Global Level

For a given language l, let $c_{l,j}$ denote the total activation count of expert j aggregated across all layers and samples. We select the top-K most frequently activated experts globally:

$$\mathcal{E}_l^{\text{global}} = \underset{j \in [N_e L]}{\arg\,\text{topK}}\ c_{l,j} \tag{1}$$

where N_e is the number of experts per layer, L is the total number of MoE layers, and K is the total number of experts considered in the statistics. To quantify the routing overlap between languages, we compute the Jaccard similarity between their top-K expert sets:

$$J(l_1, l_2) = \frac{|\mathcal{E}_{l_1}^{\text{global}} \cap \mathcal{E}_{l_2}^{\text{global}}|}{|\mathcal{E}_{l_1}^{\text{global}} \cup \mathcal{E}_{l_2}^{\text{global}}|} \tag{2}$$

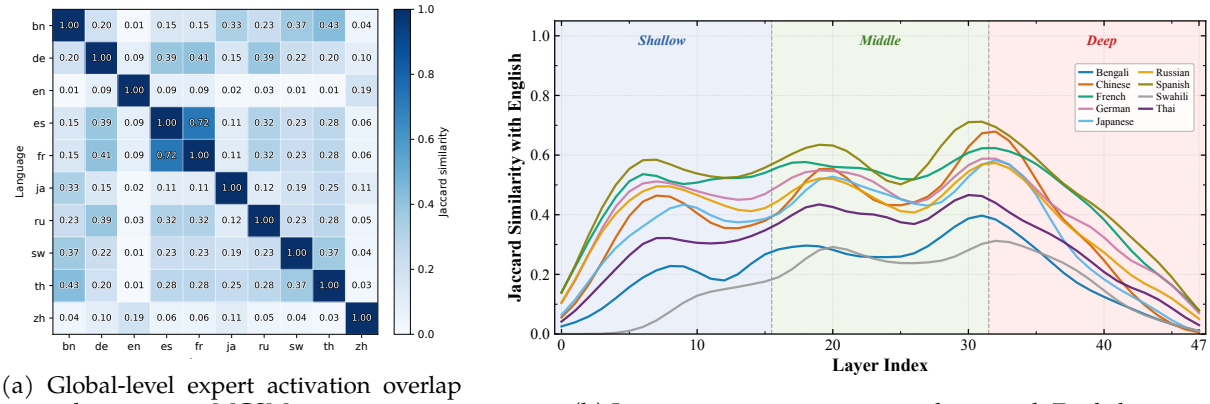
where $J(l_1, l_2) \in [0, 1]$, with 0 indicating no overlap and 1 indicating complete overlap.

In this paper, we treat English and Chinese as high-resource languages, and all other languages in the training corpus as low-resource languages, based on their relative proportions in the training data. Figure 1 (a) shows the global expert activation overlap across languages. We provide more examples of routing isolation in Appendix F.

Key observations from the global analysis point to a clear phenomenon between high- and low-resource languages: ❶ High-resource languages exhibit relatively low overlap with low-resource languages, suggesting that they follow expert-routing choices that are partly separated from those used by low-resource languages; ❷ low-resource languages show non-trivial overlap with each other, indicating that their routing choices are relatively concentrated and stable. This motivates our later attempt to select a target-language subnetwork for low-resource adaptation, and we refer to this relative separation of language-conditioned expert usage as **routing isolation**. Besides, we also found that languages from the same family (e.g., Indo-European) exhibit greater similarity in expert selection patterns, consistent with previous linguistic studies (Tang et al., 2024; Chen et al., 2026), thereby grounding our analysis in established theoretical frameworks.

3.2 *Layer-wise Convergence–Divergence* Pattern

To understand how expert routing patterns evolve across different model depths, we analyze the layer-wise expert overlap with respect to English as the reference language. For each



(a) Global-level expert activation overlap across languages in MGSM.

(b) Layer-wise expert routing similarity with English.

Figure 1: Comprehensive routing analysis of `Qwen3-30B-A3B`: global-level (left) expert activation overlap and layer-wise (right) routing similarity with English.

Language	Shallow	Middle	Deep
Bengali (BN)	0.12	0.22	0.05
Swahili (SW)	0.04	0.19	0.04
Thai (TH)	0.25	0.33	0.12
Russian (RU)	0.41	0.43	0.17

Table 2: Average layer-wise expert overlap with English. Additional results in Table 7–10.

layer i within language l, we compute the Jaccard similarity with English:

$$J_l(i) = \frac{|\mathcal{E}_l^{(i)} \cap \mathcal{E}_{\text{en}}^{(i)}|}{|\mathcal{E}_l^{(i)} \cup \mathcal{E}_{\text{en}}^{(i)}|} \quad (3)$$

This yields a per-layer similarity curve $\mathbf{J}_l = [J_l(0), \quad J_l(1), \quad \ldots, \quad J_l(L-1)]$ that captures routing dynamics across layers. The average similarity within each region $\mathcal{L} \in \{\mathcal{L}_{\text{shallow}}, \mathcal{L}_{\text{middle}}, \mathcal{L}_{\text{deep}}\}$ is:

$$\bar{J}_l^{\mathcal{L}} = \frac{1}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} J_l(i) \quad (4)$$

We plot the similarity curve $\mathbf{J}_l$ and compute the average similarity $\bar{J}_l^{\mathcal{L}}$ for each region, as shown in Figure 1 (b) and Table 2.

As shown in Table 2, routing similarity with English is consistently lower in the shallow and deep layers, while the middle layers exhibit noticeably higher overlap across all languages. This pattern suggests that the functional stratification observed in dense transformers is also reflected in MoE routing behavior. Specifically, the shallow and deep layers tend to capture more language-specific processing, whereas the middle layers appear to support more shared cross-lingual representations. These findings provide a strong empirical basis for our layer-aware expert selection strategy in RISE.

4 Method

Building on the routing isolation identified in Section 3, which suggests that MoE models contain language-specific expert subnetworks, we present a systematic approach to identify and train these subnetworks for low-resource languages, thereby improving target-language performance while minimally affecting other capabilities. Our method has three stages: (1) collecting routing statistics to understand language-expert affinities, (2) selecting language-specific and shared experts using layer-aware analysis, and (3) training the selected subnetwork while freezing the rest of the model. Figure 2 summarizes the full pipeline, and we detail each stage below.

4.1 Routing Statistics Collection

The first stage of RISE involves analyzing the routing behavior of the pre-trained model across multiple languages. This analysis reveals which experts are preferentially activated for different languages, providing the foundation for our selection strategy.

Data Preparation. We collect evaluation datasets in M languages $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_M\}$, including the target language λ^*. These datasets should cover the same or similar tasks to ensure comparable routing patterns. For each language $\lambda \in \Lambda$, we run inference on

the corresponding dataset and record the *discrete routing decisions* for all MoE layers. Specifically, let $\mathcal{R}_t^{(l)}$ denote the set of experts selected by the router at layer l for token t. We define a binary activation indicator $g_{t,i}^{(l)} = \mathbf{1}[i \in \mathcal{R}_t^{(l)}]$ and aggregate it into an activation frequency over all T_λ tokens:

$$a_{\lambda,i}^{(l)} = \frac{1}{T_\lambda} \sum_{t=1}^{T_\lambda} g_{t,i}^{(l)} \in [0,1] \tag{5}$$

where $a_{\lambda,i}^{(l)}$ represents the empirical probability that expert i is activated when processing language λ.

This yields a routing profile matrix $\mathbf{A}^{(l)} \in \mathbb{R}^{M \times N}$ for each layer, where each row corresponds to a language and each column to an expert. These matrices capture language–expert affinity patterns that inform our selection strategy.

4.2 Layer-Aware Expert Selection

Based on the analysis in Section 3, we partition the L MoE layers into three groups: shallow layers $\mathcal{L}_{\text{shallow}} = [0, L_1]$, middle layers $\mathcal{L}_{\text{middle}} = (L_1, L_2]$, and deep layers $\mathcal{L}_{\text{deep}} = (L_2, L-1]$.

Shallow and deep layers. For shallow and deep layers, we seek experts that are *preferentially activated* by the target language λ^*. We quantify this preference using the **specificity score** $S_{\lambda^*,i}^{(l)} = a_{\lambda^*,i}^{(l)} / \bar{a}_i^{(l)}$, where $\bar{a}_i^{(l)}$ is the average activation probabilities of corresponding expert i in all languages, serving as a baseline for comparison. The specificity score compares the activation probability of expert i on the target language λ^* to its average activation probability across all languages. When $S_{\lambda^*,i}^{(l)}$ is *greater than* 1, expert i is preferentially activated by λ^*, indicating language-specific behavior; when $S_{\lambda^*,i}^{(l)}$ is close to 1, the expert is activated at a similar rate across languages and thus behaves in a language-agnostic manner; and when $S_{\lambda^*,i}^{(l)}$ is *less than* 1, the expert is more preferred by other languages.

Middle layers. For middle layers, we seek experts that capture *language-agnostic* representations shared across all languages. We measure this property using the **overlap score** based on the coefficient of variation:

$$O_i^{(l)} = \frac{1}{1 + \widehat{\mathrm{c_v}}_i^{(l)}} \quad \text{where} \quad \widehat{\mathrm{c_v}}_i^{(l)} = \frac{\sigma(a_{\cdot,i}^{(l)})}{\mu(a_{\cdot,i}^{(l)})} \tag{6}$$

Here, $\widehat{c_{\mathrm{v}}}$ represents the coefficient of variation, while $\sigma(\cdot)$ and $\mu(\cdot)$ denote the standard deviation and mean of expert i's activation probabilities across all languages. The coefficient of variation measures relative dispersion:

- Low $\widehat{c_{\mathrm{v}}}$ (high $O_i^{(l)}$): Uniform activation across languages → *shared expert*.
- High $\widehat{c_{\mathrm{v}}}$ (low $O_i^{(l)}$): Activation varies significantly → *language-specific expert*.

Subnetwork Selection. The specificity and overlap scores measure relative preferences but do not account for the absolute activation magnitude. An expert with high specificity but very low overall activation may be less important than one with moderate specificity but high activation. To address this, we define composite scores that incorporate both factors:

$$\begin{aligned} \mathrm{Spec}(l,i,\lambda^*) &= S_{\lambda^*,i}^{(l)} \cdot \left(1 + \alpha \cdot a_{\lambda^*,i}^{(l)}\right), \\ \mathrm{Ovlp}(l,i) &= O_i^{(l)} \cdot \left(1 + \alpha \cdot \bar{a}_i^{(l)}\right) \end{aligned} \tag{7}$$

where $\alpha > 0$ is a hyperparameter controlling the importance of absolute activation magnitude. In our experiments, we set $\alpha = 10$.

Given a total budget of K experts to train, we allocate them across layer groups using predefined ratios (ρ_s, ρ_m, ρ_d) satisfying $\rho_s + \rho_m + \rho_d = 1$. The complete selection procedure is detailed in Algorithm 1.

With the selected expert set $\mathcal{E} = \{(l_1, i_1), \ldots, (l_K, i_K)\}$, we proceed to train only these experts while keeping all other parameters frozen.

We use standard causal language modeling loss on the target language data $\mathcal{D}_{\lambda^*}$ and our loss function is defined as:

$$\mathcal{L}(\Theta_{\text{train}}) = -\mathbb{E}_{x \sim \mathcal{D}_{\lambda^*}} \left[\sum_{t=1}^{|x|} \log P_\Theta(x_t \mid x_{<t}) \right] \tag{8}$$

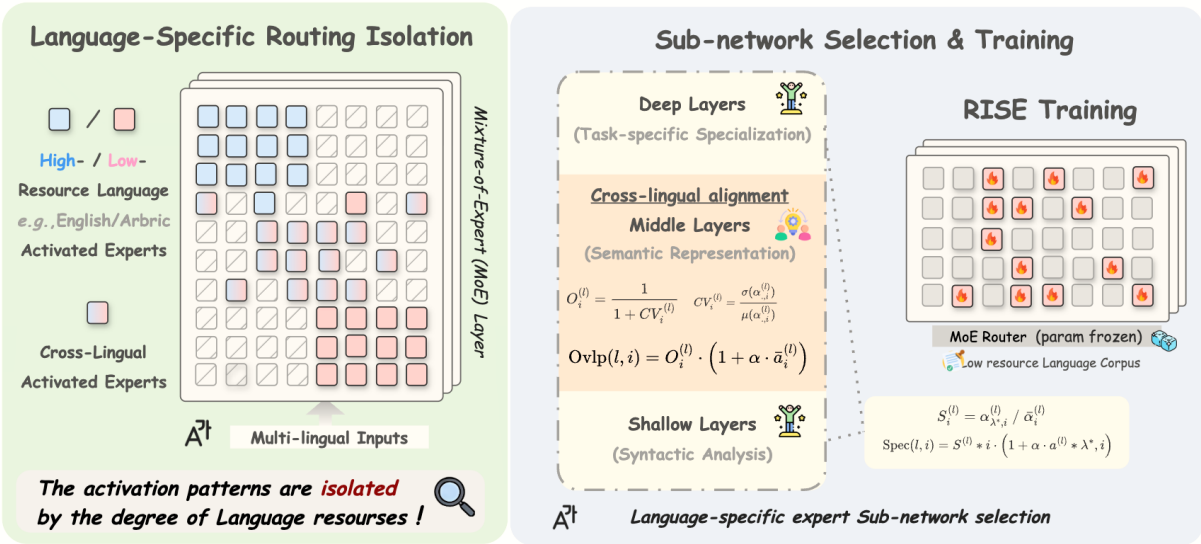


Figure 2: Overview of RISE. (a) We first collect routing statistics across multiple languages. (b) Based on layer-aware analysis, we select language-specific experts in shallow/deep layers and cross-lingual shared experts in middle layers. (c) Only the selected experts are trained.

where only Θ_{train} receives gradient updates. We detail the complete procedure in Algorithm 1 and visualize the distribution of selected experts in Appendix G.

5 Experiments

In this section, we evaluate four research questions: whether RISE improves target-language performance (RQ1), preserves non-target and general capabilities (RQ2), identifies causally important experts (RQ3), and benefits from each layer-group component (RQ4). Sensitivity analyses are deferred to Appendix E.

5.1 Settings

Datasets We evaluate on **TyDiQA-GoldP** (Clark et al., 2020), an extractive QA dataset covering 9 languages (AR, BN, EN, FI, ID, KO, RU, SW, TE), and **MGSM** (Shi et al., 2022), a math reasoning dataset covering 10 languages (BN, DE, EN, ES, FR, JA, RU, SW, TH, ZH). General capability is measured on **TriviaQA**, **MMLU**, **HellaSwag**, and **ARC**.

Comparison. We compare RISE against random expert selection, top-K activation-based selection, ESFT (Wang et al., 2024), and RISE variants that change the expert budget or target language.

Implementation Details. All experiments were conducted on a single NVIDIA H200 GPU. Training was performed for 3 epochs with a per-device batch size of 2 and gradient accumulation of 8 steps (effective batch size of 16), using a learning rate of 2×10^{-5} and `bfloat16` mixed precision. All hyperparameters are fixed across target languages and backbones; memory usage, expert allocation, and layer grouping details are in Appendix H.

5.2 Performance and Cross-lingual Preservation (RQ1 & RQ2)

To answer RQ1 and RQ2, we comprehensively compare RISE against other methods across two multilingual MoE models.

Table 3 presents the main results. We report the following observations. **Obs. ❶ RISE improves over the vanilla model and remains competitive with expert-selection baselines.** On `Qwen3-30B-A3B`, RISE achieves the best Bengali F1 of 54.23%, outperforming the other expert-selection methods in this setting. The gains also extend beyond Bengali: RISE (128, RU) and RISE (128, ID) yield the best per-language F1 on Russian (29.62%) and Indonesian (31.79%), respectively, showing that routing-based subnetwork adaptation can benefit multiple targets.

On `Phi-3.5-MoE-Instruct`, the picture is more mixed. RISE improves Bengali over Vanilla and ESFT, but TopK obtains higher

Setting	AR	BN	EN	FI	ID	KO	RU	SW	TE	Avg.
Qwen3-30B-A3B (TyDiQA-GoldP, F1 %)										
Vanilla	49.49	51.51	24.15	22.98	26.86	48.35	27.04	27.05	58.61	37.34
Random (64, BN)	48.91	**52.17**	23.87	22.31	25.61	48.19	25.63	27.73	59.03	37.05
Random (128, BN)	50.31	**53.52**	24.43	22.69	26.49	48.31	26.08	27.22	59.71	37.64
TopK (128, BN)	50.20	**52.74**	24.75	23.13	27.86	47.51	26.94	26.85	59.11	37.68
ESFT (Wang et al., 2024)	49.51	**51.79**	24.08	23.26	26.89	47.33	27.09	25.85	59.70	37.28
RISE (128, BN)	49.15	**54.23**	24.04	23.39	27.61	47.77	27.22	27.40	60.39	37.91
RISE (128, RU)	51.06	53.62	23.88	24.15	29.55	49.08	**29.62**	29.30	60.66	38.99
RISE (128, ID)	51.59	53.97	25.65	23.58	**31.79**	47.75	29.54	30.13	59.95	39.33
Phi-3.5-MoE-Instruct (TyDiQA-GoldP, F1 %)										
Vanilla	36.75	36.04	16.94	19.19	22.09	19.75	16.19	11.55	7.68	20.69
Random (16, BN)	38.26	**41.93**	18.90	19.90	23.23	26.84	18.75	11.64	9.34	23.20
Random (32, BN)	37.53	**43.61**	18.89	20.10	23.28	24.56	18.45	11.44	8.93	22.98
TopK (16, BN)	39.63	**49.51**	18.32	20.78	23.59	34.58	18.91	11.56	10.78	25.30
ESFT (Wang et al., 2024)	38.29	**45.44**	18.10	20.12	23.15	28.84	19.10	12.24	10.09	23.93
RISE (16, BN)	37.43	**46.89**	17.55	19.64	22.75	21.10	17.26	11.56	9.68	22.65
RISE (16, RU)	37.68	34.83	17.63	20.85	22.85	21.98	**17.81**	10.70	8.32	21.41
RISE (16, ID)	39.43	36.83	19.00	21.51	**22.33**	24.50	20.78	11.03	8.49	22.66

Table 3: **TyDiQA-GoldP multilingual QA performance (F1, %).** Each row corresponds to a training setting; notation (*budget, lang*) denotes the number of experts selected by RISE and the target language, e.g., (128, BN) selects 128 experts routed by Bengali and trains only those experts on Bengali data. The **colored bold** value indicates the target language column for each training setting (blue=BN, orange=RU, green=ID); the language code in the first column is colored accordingly.

Setting	TriviaQA	MMLU	HellaSwag	ARC	Avg.
Qwen	59.60	74.40	81.50	85.76	75.32
RISE	59.00	74.65	81.50	86.78	75.48
Δ	−0.60	+0.25	0.00	+1.02	+0.17
Phi	65.50	77.50	70.70	90.85	76.14
RISE	65.70	76.67	70.60	91.53	76.13
Δ	+0.20	−0.83	−0.10	+0.68	−0.01

Table 4: **General performance comparisons.** TriviaQA (world knowledge), MMLU (academic reasoning), HellaSwag (commonsense), ARC (science QA).

TyDiQA-BN F1. This suggests that, for models with a smaller expert pool and different routing sparsity, frequency-based expert tuning can already capture a strong target-language path; we therefore view the Phi result as architecture-dependent behavior rather than universal dominance over simpler expert-selection rules.

Obs. ❷ RISE delivers targeted improvements without cross-lingual or cross-task degradation. Table 6 shows that RISE trained only on TyDiQA preserves held-out MGSM performance: RISE (128, BN) reaches 80.7% average accuracy, close to Vanilla (80.2%). This suggests that the selected subnetworks capture language-specific rather than task-general computation. Table 4 further shows that general-capability shifts are bounded within 1.02 points, indicating that selective expert updates leave shared, task-agnostic computation largely intact.

5.3 Causal Expert Verification (RQ3)

Obs. ❸ The RISE-selected experts are causally responsible for target-language computation. To verify that the identified subnet is genuinely language-specific, we prune the selected Bengali subnetwork and re-evaluate both backbones on TyDiQA-GoldP (Table 5). Bengali suffers catastrophic collapse, while most non-target languages decline more moderately, mirroring the selectivity observed during training. This bidirectional evidence, improvement upon activation and collapse upon ablation, supports that RISE identifies a functionally important target-language subnetwork rather than a task-general or spuriously correlated set of experts.

5.4 Indispensability of Layer-Group Components (RQ4)

We train RISE variants that omit or retain individual layer groups (Figure 3). **Obs. ❹ All three layer groups contribute to RISE,**

Setting	AR	BN	EN	FI	ID	KO	RU	SW	TE
Qwen3-30B-A3B (TyDiQA-GoldP, F1 %)									
Vanilla	49.49	51.51	24.15	22.98	26.86	48.35	27.04	27.05	58.61
RISE (128, BN)	49.15	**54.23**	24.04	23.39	27.61	47.77	27.22	27.40	60.39
Pruned (128, BN)	26.83$_{\downarrow 22.32}$	**1.00**$_{\downarrow 53.23}$	24.63$_{\uparrow 0.59}$	19.37$_{\downarrow 4.02}$	30.99$_{\uparrow 3.38}$	12.47$_{\downarrow 35.30}$	17.87$_{\downarrow 9.35}$	21.09$_{\downarrow 6.31}$	7.78$_{\downarrow 52.61}$
Phi-3.5-MoE-Instruct (TyDiQA-GoldP, F1 %)									
Vanilla	36.75	36.04	16.94	19.19	22.09	19.75	16.19	11.55	7.68
RISE (16, BN)	37.43	**46.89**	17.55	19.64	22.75	21.10	17.26	11.56	9.68
Pruned (16, BN)	12.03$_{\downarrow 25.40}$	**1.85**$_{\downarrow 45.04}$	16.53$_{\downarrow 1.02}$	19.40$_{\downarrow 0.24}$	18.79$_{\downarrow 3.96}$	16.63$_{\downarrow 4.47}$	7.85$_{\downarrow 9.41}$	3.46$_{\downarrow 8.10}$	1.02$_{\downarrow 8.66}$

Table 5: **Causal expert verification on TyDiQA-GoldP (F1, %).** **Colored bold**: target-language column. Pruned: Bengali expert subnetwork ablated.

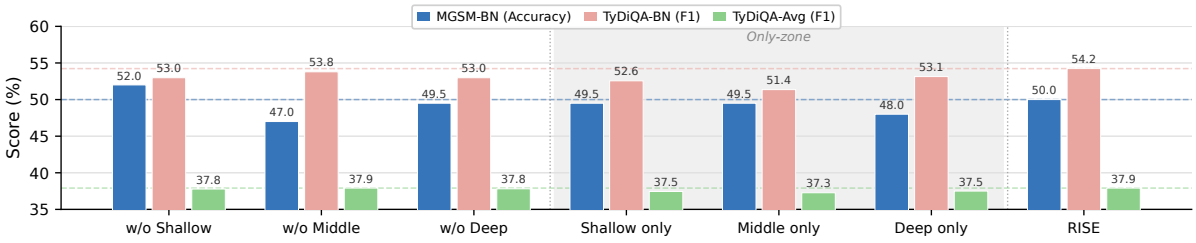


Figure 3: **Ablation of layer-wise expert groups.** *w/o*: remove layer group; *Only*: retain only that group.

with shallow and deep layers being most critical for target-language adaptation. Removing shallow or deep experts causes the largest TyDiQA-BN drops, whereas *w/o Middle* degrades least, consistent with middle-layer routing being more language-agnostic. By contrast, experts selected in shallow and deep layers capture more language-specific processing that is essential for low-resource adaptation. No single group alone matches the full method, confirming that the layer-aware design is non-redundant and that the three groups play complementary roles.

Qualitative Case Study. Appendix J presents four qualitative cases comparing the Vanilla baseline and RISE on Bengali MGSM and TyDiQA-GoldP. Two recurring failure modes are identified: mathematical reasoning errors and language coherence failures. RISE corrects both categories, suggesting that activating the Bengali-specific expert subnetwork improves not only aggregate F1 but also output coherence.

6 Conclusion

We identify **language routing isolation** in multilingual MoE models, where high- and low-resource languages tend to follow distinct expert-routing pathways. Building on this mechanism, we propose RISE, a layer-aware framework that selects and trains target-adaptive expert subnetworks while freezing the rest of the model. Experiments on TyDiQA and MGSM show that RISE improves low-resource performance while preserving non-target and general capabilities, demonstrating that routing-level interpretability can directly support efficient multilingual adaptation. Overall, these results show that routing statistics can serve not only as a diagnostic tool for multilingual MoE behavior, but also as a practical guide for selective adaptation under tight parameter budgets.

Limitations

Our study is limited to two multilingual MoE backbones due to the limitations of multilingual MoE models and two main multilingual benchmarks, so the observed routing patterns and adaptation behavior may not generalize to all MoE architectures, languages, or domains. In addition, routing statistics depend on the profiling data, prompt format, tokenization, and task distribution; so it may be affected by the distributions of domain a littel. Finally, RISE assumes that the pretrained MoE model already contains a useful target-language expert footprint, and may be less effective for some really severely underrepresented languages.

References

Marah Abdin and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219.

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. 2026. Multilingual routing in mixture-of-experts. In *International Conference on Learning Representations (ICLR) 2026*. Accepted paper / OpenReview submission.

Ryo Bertolissi, Jonas Hübotter, Ido Hakimi, and Andreas Krause. 2025. Local mixtures of experts: Essentially free test-time training via model merging. In *Conference on Language Modeling (COLM)*.

Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte Miguel Alves, Andre Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El Haddad, Manuel Faysse, Maxime Peyrard, Nuno M Guerreiro, Patrick Fernandes, Ricardo Rei, and Pierre Colombo. 2025. EuroBERT: Scaling multilingual encoders for european languages. In *Conference on Language Modeling (COLM)*.

Marmik Chaudhari, Idhant Gulati, Nishkal Hundia, Pranav Karra, and Shivam Raval. 2026. Moe lens – an expert is all you need. *Preprint*, arXiv:2603.05806.

Yuxin Chen, Zhengzhou Cai, Xiangtian Ji, Weixiang Zhao, An Zhang, Xiang Wang, and Tat-Seng Chua. 2026. Understanding multilingualism in mixture-of-experts llms: Routing mechanism, expert specialization, and layerwise steering. *Preprint*, arXiv:2601.14050.

Ikhyun Cho and Julia Hockenmaier. 2025. Analyzing multilingualism in large language models with sparse autoencoders. In *Second Conference on Language Modeling*.

Lynn Chua, Badih Ghazi, Yangsibo Huang, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, Chulin Xie, and Chiyuan Zhang. 2025. Crosslingual capabilities and knowledge barriers in multilingual large language models. *Preprint*, arXiv:2406.16135.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Preprint*, arXiv:2003.05002.

DeepSeek-AI and 1 others. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434. 2024a.

DeepSeek-AI and 1 others. 2024b. Deepseek-v3 technical report. *CoRR*, abs/2412.19437. 2024b.

Guanzhi Deng, Bo Li, Ronghao Chen, Huacan Wang, Lijie Wen, and Linqi Song. 2026. Dr-lora: Dynamic rank lora for mixture-of-experts adaptation. *Preprint*, arXiv:2601.04823.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

William Fleshman and Benjamin Van Durme. 2025. SpectR: Dynamically composing LM experts with spectral routing. In *Conference on Language Modeling (COLM)*.

Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Xinye Cao, Sicong Leng, Qimei Cui, and Xudong Jiang. 2026. Advancing expert specialization for better moe. *Preprint*, arXiv:2505.22323.

Yekyung Kim, Jenna Russell, Marzena Karpinska, and Mohit Iyyer. 2025. One ruler to measure them all: Benchmarking multilingual long-context language models. In *Conference on Language Modeling (COLM)*.

Priyanshu Kumar, Devansh Jain, Akhila Yerukola, Liwei Jiang, Himanshu Beniwal, Thomas Hartvigsen, and Maarten Sap. 2025. PolyGuard: A multilingual safety moderation tool for 17 languages. In *Conference on Language Modeling (COLM)*.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *Preprint*, arXiv:2006.16668.

Junzhuo Li, Bo Wang, Xiuze Zhou, and Xuming Hu. 2025a. Dynamic expert specialization: Towards catastrophic forgetting-free multi-domain MoE adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18478–18493, Suzhou, China. Association for Computational Linguistics.

Zhongyang Li, Ziyue Li, and Tianyi Zhou. 2025b. C3PO: Critical-layer, core-expert, collaborative pathway optimization for test-time expert remixing. In *Conference on Language Modeling (COLM)*.

Zichong Li, Chen Liang, Zixuan Zhang, Ilgee Hong, Young Jin Kim, Weizhu Chen, and Tuo Zhao. 2025c. SlimMoE: Structured compression of large MoE models via expert slimming and distillation. In *Conference on Language Modeling (COLM)*.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. FineWeb2: One pipeline to scale them all — adapting pre-training data processing to every language. In *Conference on Language Modeling* (*COLM*).

José Pombal, Dongkeun Yoon, Patrick Fernandes, Ian Wu, Seungone Kim, Ricardo Rei, Graham Neubig, and André Martins. 2025. M-Prometheus: A suite of open multilingual LLM judges. In *Conference on Language Modeling* (*COLM*).

Miguel Moura Ramos, Patrick Fernandes, Sweta Agrawal, and Andre Martins. 2025. Multilingual contextualization of large language models for document-level machine translation. In *Conference on Language Modeling* (*COLM*).

Fabian David Schmidt, Ivan Vulić, Goran Glavaš, and David Ifeoluwa Adelani. 2025. Fleurs-SLU: A massively multilingual benchmark for spoken language understanding. In *Conference on Language Modeling* (*COLM*).

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. *Preprint*, arXiv:1701.06538.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *Preprint*, arXiv:2210.03057.

Chenyang Song, Weilin Zhao, Xu Han, Chaojun Xiao, Yingfa Chen, Yuxuan Li, Zhiyuan Liu, and Maosong Sun. 2025. BlockFFN: Towards end-side acceleration-friendly mixture-of-experts with chunk-level activation sparsity. In *Conference on Language Modeling* (*COLM*).

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Y. Wu. 2024. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models. *Preprint*, arXiv:2407.01906.

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

Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel F Campos. 2025. Arctic-Embed 2.0: Multilingual retrieval without compromise. In *Conference on Language Modeling* (*COLM*).

Xinyu Zhao, Xuxi Chen, Yu Cheng, and Tianlong Chen. 2024. Sparse moe with language guided routing for multilingual machine translation. In *The Twelfth International Conference on Learning Representations*.

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. *Preprint*, arXiv:2410.10626.

Appendix

A Additional Related Work **11**

B The Algorithm of RISE **12**

C Gradient Isolation **12**
- C.1 Gradient Isolation 13
- C.2 Exact Preservation Under Disjoint Routing 13
- C.3 Approximate Preservation Under Near-Orthogonal Routing . 14
- C.4 Implication for RISE 15

D Cross-Task Transfer: MGSM Results After TyDiQA Training **15**

E Additional Sensitivity Analysis **15**

F Additional Routing Analysis **16**

G RISE Selected Expert Visualization **17**

H Layer Partition and Expert Selection Settings **18**

I Why We Do Not Compare with LoRA **18**

J Case Study **19**

A Additional Related Work

Multilingual Language Models As large language models have gained increasing prominence, a growing body of research has begun to focus on both the mechanistic interpretability and practical applications of multilingual language models. Cho and Hockenmaier (2025) applied sparse autoencoders to dissect multilingual processing in LLMs, revealing how individual features encode language-specific information. On the data side, Penedo et al. (2025) introduced FineWeb2, a scalable pre-training data pipeline adapted to every language, enabling the development of high-quality multilingual corpora at scale. For multilingual encoders, Boizard et al. (2025) proposed EuroBERT, scaling multilingual encoder models across European languages, while Yu et al. (2025) presented Arctic-Embed 2.0, achieving strong multilingual retrieval without sacrificing cross-lingual transfer. Benchmarking efforts have also expanded in scope: Kim et al. (2025) introduced a benchmark for multilingual long-context language models, and Schmidt et al. (2025) proposed Fleurs-SLU, a massively multilingual benchmark for spoken language understanding. On the evaluation and judgment side, Pombal et al. (2025) developed M-Prometheus, a suite of open multilingual LLM judges supporting diverse languages. For safety, Kumar et al. (2025) introduced PolyGuard, a multilingual moderation tool covering 17 languages. Regarding translation, Ramos et al. (2025) studied multilingual contextualization of LLMs for document-level machine translation. Collectively, these works highlight the growing importance of multilingual considerations across all stages of language model development, providing important context for our analysis of multilingual routing behavior in MoE models.

Mixture-of-Experts (MoE) Mixture-of-Experts (MoE) architectures have been widely studied as a means to scale neural networks while maintaining computational efficiency (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022). By activating only a subset of experts for each input token, MoE models can achieve high parameter counts without incurring proportional increases in inference cost. This design has been particularly influential in the development of large language models (LLMs), where MoE variants have demonstrated strong performance across various natural language processing tasks (DeepSeek-AI et al., 2024a; Abdin et al., 2024; Yang et al., 2025). A key component of MoE models is the routing mechanism, which determines how input tokens are assigned to different experts. Several routing strategies have been proposed, including top-k gating (Shazeer et al., 2017), learned routing networks (Lepikhin et al., 2020), and dynamic routing based on input features (Li et al., 2025a). Beyond standard routing, recent work has explored more flexible expert composition strategies: Fleshman and Durme (2025) proposed SpectR, which dynamically composes experts using spectral routing to improve model expressivity, while Li et al. (2025b) introduced C3PO, a test-time expert re-mixing approach that optimizes expert

pathways at critical layers without retraining. Complementary efforts have focused on efficiency and compression: Li et al. (2025c) proposed SlimMoE to compress large MoE models via expert slimming and distillation, Song et al. (2025) introduced BlockFFN with chunk-level activation sparsity to accelerate inference on end-side devices, and Bertolissi et al. (2025) showed that local mixtures of experts can be constructed via model merging to enable essentially free test-time training.

B The Algorithm of RISE

Algorithm 1 Language-specific Expert Subnetwork Selection

Require: Routing profiles $\{\mathbf{A}^{(l)}\}_{l=0}^{L-1}$, target language λ^*, budget K, layer boundaries (L_1, L_2), allocation ratios (ρ_s, ρ_m, ρ_d)

Ensure: Selected expert set $\mathcal{E}$

Initialize: $\mathcal{E} \leftarrow \emptyset$

Compute budgets: $K_s \leftarrow \lfloor K \cdot \rho_s \rfloor$, $K_m \leftarrow \lfloor K \cdot \rho_m \rfloor$, $K_d \leftarrow K - K_s - K_m$

// Phase 1: Select language-specific experts from shallow layers

for each layer $l \in \mathcal{L}_{\text{shallow}}$ and expert $i \in [N]$ **do**

  Compute $\text{Spec}(l, i, \lambda^*)$ using Eq. (7)

end for

$\mathcal{E} \leftarrow \mathcal{E} \cup \text{TopK}\left(\{(l,i) : l \in \mathcal{L}_{\text{shallow}}\}, K_s\right)$

// Phase 2: Select cross-lingual shared experts from middle layers

for each layer $l \in \mathcal{L}_{\text{middle}}$ and expert $i \in [N]$ **do**

  Compute $\text{Ovlp}(l, i)$ using Eq. (6)

end for

$\mathcal{E} \leftarrow \mathcal{E} \cup \text{TopK}\left(\{(l,i) : l \in \mathcal{L}_{\text{middle}}, (l,i) \notin \mathcal{E}\}, K_m\right)$

// Phase 3: Select language-specific experts from deep layers

for each layer $l \in \mathcal{L}_{\text{deep}}$ and expert $i \in [N]$ **do**

  Compute $\text{Spec}(l, i, \lambda^*)$ using Eq. (7)

end for

$\mathcal{E} \leftarrow \mathcal{E} \cup \text{TopK}\left(\{(l,i) : l \in \mathcal{L}_{\text{deep}}, (l,i) \notin \mathcal{E}\}, K_d\right)$

return $\mathcal{E}$

C Gradient Isolation

In this appendix, we formalize why routing isolation leads to gradient isolation, and why training only the target-language expert subnetwork preserves the model's general capabilities. The key point is that sparse MoE routing does not merely reduce computation: it also induces a structured partition of the parameter space. Once the routing supports of different languages are separated, gradient updates become correspondingly localized.

Setup. Consider an MoE transformer with L MoE layers and N_e experts per layer. For an input sequence $x = (x_1, \ldots, x_T)$, let $h_t^{(l)}$ denote the hidden state of token t at layer l. At MoE layer l, the output is

$$h_t^{(l+1)} = U^{(l)}\left(h_t^{(l)}\right) + \sum_{i=1}^{N_e} g_{t,i}^{(l)}(x)\, F_i^{(l)}\left(h_t^{(l)}; \theta_{l,i}\right), \tag{9}$$

where: (i) $U^{(l)}$ collects all frozen shared transformations at layer l (e.g., residual pathway, attention block, layer norm, and any non-expert shared modules), (ii) $F_i^{(l)}(\cdot; \theta_{l,i})$ is expert i at layer l with parameters $\theta_{l,i}$, and (iii) $g_{t,i}^{(l)}(x) \in \{0, 1\}$ is the discrete routing indicator, with $\sum_{i=1}^{N_e} g_{t,i}^{(l)}(x) = k$ under top-k routing.

Let $S = \{S_l\}_{l=1}^{L}$ denote the selected expert subnetwork, where $S_l \subseteq \{1, \ldots, N_e\}$ is the set of trainable experts in layer l. During RISE training, only $\theta_S = \{\theta_{l,i} : i \in S_l\}$ are updated; all other parameters are frozen.

For a language λ, define its routing support at layer l as

$$\mathcal{R}_\lambda^{(l)} = \Big\{ i \in \{1, \ldots, N_e\} : \Pr_{x \sim \mathcal{D}_\lambda,\, t}\left[g_{t,i}^{(l)}(x) = 1\right] > 0 \Big\}. \tag{10}$$

Intuitively, $\mathcal{R}_\lambda^{(l)}$ contains the experts that language λ actually uses at layer l.

We also define the routing overlap mass between language λ and the selected subnetwork S:

$$\Omega_\lambda(S) = \sum_{l=1}^{L} \sum_{i \in S_l} p_{\lambda,i}^{(l)}, \qquad p_{\lambda,i}^{(l)} = \mathbb{E}_{x \sim \mathcal{D}_\lambda,\, t}\left[g_{t,i}^{(l)}(x)\right]. \tag{11}$$

When routing isolation is strong, $\Omega_\lambda(S)$ is small for non-target languages.

C.1 Gradient Isolation

We first state the exact gradient form for expert parameters.

Lemma 1 (Exact gradient isolation)**.** *For any expert i at layer l, the gradient of the training loss $\mathcal{L}_{\lambda^*}$ on target-language data $\mathcal{D}_{\lambda^*}$ satisfies*

$$\nabla_{\theta_{l,i}} \mathcal{L}_{\lambda^*} = \mathbb{E}_{x \sim \mathcal{D}_{\lambda^*}} \left[\sum_{t=1}^{T} g_{t,i}^{(l)}(x) \, \nabla_{\theta_{l,i}} \ell_t(x) \right], \tag{12}$$

where $\ell_t(x)$ is the token-level loss contribution. Hence, if expert i is never activated by target-language tokens, i.e., $g_{t,i}^{(l)}(x) = 0$ for all (x, t) from $\mathcal{D}_{\lambda^*}$, then

$$\nabla_{\theta_{l,i}} \mathcal{L}_{\lambda^*} = 0. \tag{13}$$

Proof. From the MoE forward definition in Eq. (9), the parameter $\theta_{l,i}$ appears only in the term $g_{t,i}^{(l)}(x) F_i^{(l)}(h_t^{(l)}; \theta_{l,i})$. Since $g_{t,i}^{(l)}(x) \in \{0, 1\}$ is a multiplicative routing mask, the chain rule gives

$$\nabla_{\theta_{l,i}} \ell_t(x) = g_{t,i}^{(l)}(x) \, \frac{\partial \ell_t(x)}{\partial F_i^{(l)}} \, \frac{\partial F_i^{(l)}(h_t^{(l)}; \theta_{l,i})}{\partial \theta_{l,i}}. \tag{14}$$

Summing over tokens and taking expectation over $x \sim \mathcal{D}_{\lambda^*}$ yields Eq. (12). If $g_{t,i}^{(l)}(x) = 0$ for all target-language examples, every term vanishes identically, so the gradient is exactly zero. □

This lemma shows that sparse routing induces exact gradient sparsity: an expert receives gradient only from the tokens that actually traverse it.

C.2 Exact Preservation Under Disjoint Routing

We now formalize the strongest case: if the selected target-language subnetwork is disjoint from the routing support of another language, then training on the target language leaves that other language *exactly unchanged*.

Theorem 1 (Exact invariance under disjoint routing)**.** *Let λ^* be the target language and let $S = \{S_l\}_{l=1}^{L}$ be the selected trainable expert subnetwork. Assume:*

1. *All shared parameters and all unselected experts are frozen.*
2. *For a non-target language λ, the selected experts are disjoint from its routing supports at every layer:*

$$S_l \cap \mathcal{R}_\lambda^{(l)} = \varnothing, \qquad \forall l = 1, \ldots, L. \tag{15}$$

Then, after any number of gradient-based training steps on $\mathcal{D}_{\lambda^*}$ that update only θ_S, the model's forward computation on any input $x \sim \mathcal{D}_\lambda$ remains exactly unchanged:

$$f_{\theta'}(x) = f_\theta(x), \qquad \forall x \in \mathrm{supp}(\mathcal{D}_\lambda), \tag{16}$$

where θ denotes the initial parameters and θ' the parameters after training. Consequently, the predictive distribution, token losses, and task outputs on language λ are all unchanged.

Proof. We prove the claim by induction over training steps and model layers.

Step 1: one update cannot alter the forward path of language λ. Consider one optimization step that updates only θ_S:

$$\theta'_{l,i} = \begin{cases} \theta_{l,i} - \eta \nabla_{\theta_{l,i}} \mathcal{L}_{\lambda^*}, & i \in S_l, \\ \theta_{l,i}, & i \notin S_l. \end{cases} \tag{17}$$

Take any input $x \sim \mathcal{D}_\lambda$. We show by induction on layer depth that all hidden states remain unchanged.

For the first layer, the input embeddings are unchanged because they are frozen, so $h_t^{(0)\prime} = h_t^{(0)}$.

Assume $h_t^{(l)\prime} = h_t^{(l)}$ for all tokens t at layer l. Since the router and all shared modules are frozen, the router receives the same input hidden states before and after the update; therefore the routing decisions are identical:

$$g_{t,i}^{(l)\prime}(x) = g_{t,i}^{(l)}(x), \qquad \forall i, t. \tag{18}$$

By the disjointness assumption in Eq. (15), if $g_{t,i}^{(l)}(x) = 1$, then necessarily $i \notin S_l$, because no selected expert belongs to the routing support of language λ. Hence every expert actually activated by language λ at layer l remains frozen:

$$\theta'_{l,i} = \theta_{l,i} \quad \text{for all } i \text{ such that } g_{t,i}^{(l)}(x) = 1. \tag{19}$$

Therefore, every active expert output is unchanged, and so is the shared branch:

$$h_t^{(l+1)\prime} = U^{(l)}(h_t^{(l)\prime}) + \sum_{i=1}^{N_e} g_{t,i}^{(l)\prime}(x)\, F_i^{(l)}(h_t^{(l)\prime}; \theta'_{l,i}) \tag{20}$$

$$= U^{(l)}(h_t^{(l)}) + \sum_{i=1}^{N_e} g_{t,i}^{(l)}(x)\, F_i^{(l)}(h_t^{(l)}; \theta_{l,i}) \tag{21}$$

$$= h_t^{(l+1)}. \tag{22}$$

Thus the induction closes, giving $h_t^{(l)\prime} = h_t^{(l)}$ for all layers l and tokens t, and therefore $f_{\theta'}(x) = f_\theta(x)$.

Step 2: extension to multiple updates. The argument above applies after each gradient step individually. Since at every step only selected experts are modified, and these experts are never traversed by language λ, the forward computation for λ remains unchanged after every step. By induction over optimization steps, Eq. (16) holds after arbitrary-length training. □

Theorem 1 gives the cleanest formal statement of the intuition: if a language never uses the updated experts, then it is mathematically impossible for its forward pass to change.

C.3 Approximate Preservation Under Near-Orthogonal Routing

In practice, routing isolation is strong but not perfectly disjoint. We therefore derive a stability bound showing that cross-lingual interference is controlled by routing overlap.

Assumption 1 (Expert smoothness)**.** *For each expert $F_i^{(l)}$, there exists a constant $L_{l,i} > 0$ such that for all relevant hidden states h and parameter perturbations $\Delta\theta_{l,i}$,*

$$\left\| F_i^{(l)}(h; \theta_{l,i} + \Delta\theta_{l,i}) - F_i^{(l)}(h; \theta_{l,i}) \right\| \le L_{l,i}\, \|\Delta\theta_{l,i}\|. \tag{23}$$

Assumption 2 (Task-head smoothness)**.** *The mapping from final hidden states to logits/loss is C-Lipschitz.*

Theorem 2 (Cross-lingual perturbation bound)**.** *Let $\Delta\theta$ be any update supported only on the selected subnetwork S, i.e., $\Delta\theta_{l,i} = 0$ for $i \notin S_l$. Assume the routing pattern of a non-target language λ is locally stable under this small update. Then the expected output perturbation on $\mathcal{D}_\lambda$ is bounded by*

$$\mathbb{E}_{x\sim\mathcal{D}_\lambda}\left[\|f_{\theta+\Delta\theta}(x) - f_\theta(x)\|\right] \le C \sum_{l=1}^{L} \sum_{i\in S_l} p_{\lambda,i}^{(l)}\, L_{l,i}\, \|\Delta\theta_{l,i}\|. \tag{24}$$

Consequently, the expected loss change satisfies

$$\mathbb{E}_{x\sim\mathcal{D}_\lambda}\left[|\ell(x;\theta+\Delta\theta) - \ell(x;\theta)|\right] \le C' \sum_{l=1}^{L} \sum_{i\in S_l} p_{\lambda,i}^{(l)}\, L_{l,i}\, \|\Delta\theta_{l,i}\| \tag{25}$$

for some constant $C' > 0$.

Proof. Fix $x \sim \mathcal{D}_\lambda$ and condition on its routing pattern. Because only experts in S are updated, the change in the MoE contribution at layer l can only come from activated experts in S_l. Using Eq. (9), define the per-expert perturbation

$$\Delta F_{t,i}^{(l)} := F_i^{(l)}(h_t^{(l)}; \theta_{l,i} + \Delta\theta_{l,i}) - F_i^{(l)}(h_t^{(l)}; \theta_{l,i}).$$

The triangle inequality then gives

$$\|h_t^{(l+1)\prime} - h_t^{(l+1)}\| \le \sum_{i\in S_l} g_{t,i}^{(l)}(x)\, \|\Delta F_{t,i}^{(l)}\|. \tag{26}$$

Applying the smoothness assumption in Eq. (23) gives

$$\|h_t^{(l+1)\prime} - h_t^{(l+1)}\| \le \sum_{i\in S_l} g_{t,i}^{(l)}(x)\, L_{l,i}\, \|\Delta\theta_{l,i}\|. \tag{27}$$

Summing across layers and propagating to the output using the C-Lipschitz property of the remaining frozen network yields

$$\|f_{\theta+\Delta\theta}(x) - f_\theta(x)\| \le C \sum_{l=1}^{L} \sum_{i\in S_l} g_{t,i}^{(l)}(x)\, L_{l,i}\, \|\Delta\theta_{l,i}\|. \tag{28}$$

Taking expectation over $x \sim \mathcal{D}_\lambda$ and using $\mathbb{E}[g_{t,i}^{(l)}(x)] = p_{\lambda,i}^{(l)}$ gives Eq. (24). The loss

bound in Eq. (25) follows immediately from the Lipschitz continuity of the loss with respect to logits. □

Theorem 2 shows that cross-lingual interference is not arbitrary: it is proportional to the extent to which a non-target language routes through the updated experts. Therefore, when routing isolation makes $p^{(l)}_{\lambda,i}$ very small for $i \in S_l$, the induced perturbation on language λ is correspondingly small.

C.4 Implication for RISE

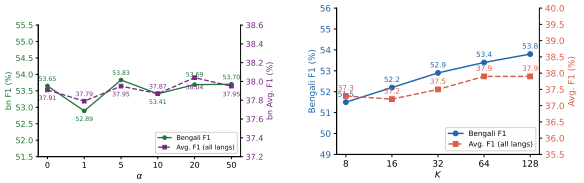
The above results justify the design principle of RISE. The method first identifies a target-language expert subnetwork using routing statistics, and then updates only that subnetwork while freezing all remaining parameters. If the selected experts align with the target language's routing support and have little overlap with other languages, then:

1. gradients from target-language training are concentrated on the selected experts;
2. non-target languages do not backpropagate through those experts, or do so only with very small probability;
3. thus, non-target behavior is either exactly preserved (in the disjoint case) or perturbed only by a small amount bounded by routing overlap.

In this sense, routing isolation induces *functional decoupling* between language-specific expert subnetworks, while gradient isolation turns this decoupling into *optimization locality*. This explains why selective expert training can improve the target language without materially degrading the model's general multilingual capabilities.

D Cross-Task Transfer: MGSM Results After TyDiQA Training

To further validate that RISE does not impair the model's capabilities on tasks beyond the training domain, we evaluate all TyDiQA-trained models on the MGSM benchmark without any MGSM-specific training. This cross-task evaluation directly tests whether the language-specific subnetworks identified by RISE are truly isolated: training on extractive QA (TyDiQA) should not degrade multilingual mathematical reasoning (MGSM) if the selected experts are genuinely language-specific rather than task-general. As shown in Table 6, RISE consistently preserves MGSM performance across all non-target languages, confirming that our subnetwork selection respects the functional boundaries between language-specific and task-general computation.



(a) Scale factor α. (b) Selected experts K.

Figure 4: Sensitivity analysis for two key hyperparameters: **(a)** activation scale factor α and **(b)** total number of selected experts K.

E Additional Sensitivity Analysis

As shown in Figure 4(a), performance is stable across a broad range of α values, with a modest optimum that reflects the intended balance between activation frequency and routing weight in the composite selection score. Figure 4(b) demonstrates that performance improves as K increases up to 128, beyond which returns diminish. Figure 5 further varies the layer-group budget allocation ratio across shallow, middle, and deep experts. The best setting concentrates the budget on shallow and deep layers, while the uniform allocation underperforms, matching the layer-wise ablation in Figure 3.

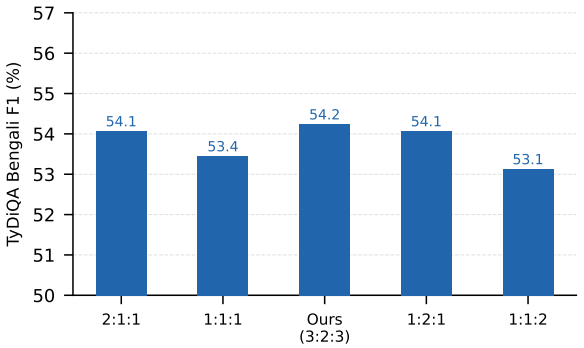


Figure 5: Ablation on expert budget allocation ratio across shallow, middle, and deep layers.

Setting	BN	DE	EN	ES	FR	JA	RU	SW	TH	ZH	Avg.
Qwen3-30B-A3B (MGSM, Accuracy %)											
Vanilla	46.0	88.5	96.5	91.5	82.0	83.5	92.5	48.0	87.5	86.0	80.2
Random (64,BN)	**46.0**	87.5	96.0	90.5	81.5	84.5	91.0	46.5	87.0	89.0	80.0
Random (128,BN)	**49.0**	87.5	96.0	90.0	81.5	83.0	91.0	46.0	87.0	87.0	79.8
ESFT (Wang et al., 2024)	**48.0**	86.5	97.0	91.0	82.0	84.0	92.5	47.5	87.0	86.0	80.2
RISE (128, BN)	**55.0**	87.5	96.5	89.5	82.5	84.5	91.0	46.5	86.0	87.5	80.7
RISE (128, RU)	49.0	88.5	96.0	91.5	81.0	85.0	**91.0**	43.0	87.5	89.0	80.2
RISE (128, ID)	46.5	88.5	97.5	90.5	82.0	84.5	92.0	47.0	88.0	87.0	80.4
Phi-3.5-MoE-Instruct (vLLM + chat template, MGSM, Accuracy %)											
Vanilla	1.0	79.0	88.5	81.5	73.0	56.0	77.0	1.0	26.5	65.5	54.9
Random (16,BN)	**3.0**	79.5	89.5	83.0	79.0	60.0	82.0	1.5	25.0	65.5	56.8
Random (32,BN)	**3.0**	78.5	88.0	83.5	79.5	57.0	78.5	1.5	22.5	64.5	55.7
TopK (16,BN)	**1.5**	78.5	89.0	82.0	74.0	59.5	79.0	2.5	24.5	67.5	55.8
ESFT (Wang et al., 2024)	**3.5**	81.0	88.5	82.5	78.0	56.5	80.5	1.5	22.5	66.0	56.1
RISE (16, BN)	**3.5**	76.5	88.5	84.0	75.5	62.5	81.5	0.5	22.0	64.0	55.9
RISE (16, RU)	2.5	79.5	91.0	83.5	75.5	58.0	**79.0**	0.0	19.5	66.0	55.5
RISE (16, ID)	0.5	80.5	88.5	83.5	76.5	58.5	78.0	1.5	19.5	63.5	55.1

Table 6: **Cross-task transfer: MGSM accuracy (%) for models trained on TyDiQA.** All models are trained exclusively on TyDiQA and evaluated zero-shot on MGSM to measure cross-task interference. The table is split by backbone: `Qwen3-30B-A3B` and `Phi-3.5-MoE-Instruct`. The **colored bold** value indicates the target language column (blue=BN, orange=RU, green=ID); the language code in the first column is colored accordingly.

Language	Shal.	Mid.	Deep	Avg.
Telugu (TE)	0.16	0.29	0.06	0.17
Swahili (SW)	0.14	0.30	0.07	0.17
Bengali (BN)	0.19	0.31	0.10	0.20
Korean (KO)	0.27	0.39	0.12	0.26
Finnish (FI)	0.29	0.41	0.14	0.28
Arabic (AR)	0.31	0.42	0.21	0.31
Indonesian (ID)	0.32	0.46	0.19	0.32
Russian (RU)	0.40	0.42	0.21	0.34
Mean	**0.26**	**0.38**	**0.14**	**0.26**

Table 7: Layer-wise expert overlap with English on TyDiQA (Qwen3-30B-A3B).

Language	Shal.	Mid.	Deep	Avg.
Telugu (TE)	0.14	0.12	0.06	0.11
Bengali (BN)	0.24	0.16	0.00	0.13
Arabic (AR)	0.19	0.23	0.00	0.14
Swahili (SW)	0.19	0.19	0.11	0.16
Finnish (FI)	0.24	0.23	0.06	0.17
Korean (KO)	0.24	0.19	0.11	0.18
Indonesian (ID)	0.29	0.26	0.06	0.20
Russian (RU)	0.33	0.39	0.00	0.24
Mean	**0.23**	**0.22**	**0.05**	**0.17**

Table 8: Layer-wise expert overlap with English on TyDiQA (Phi-3.5-MoE-Instruct).

F Additional Routing Analysis

This section consolidates the complete layer-wise expert overlap statistics across both backbone models and both evaluation benchmarks, extending the analysis summarized in Table 2 of the main text.

Qwen3-30B-A3B. Tables 7 and 9 report per-language Jaccard similarity with English at the shallow, middle, and deep layers for `Qwen3-30B-A3B` on TyDiQA and MGSM, respectively. Besides, we also provide the global-level TyDiQA overlap heatmap in Figure 6, which supplements Figure 1 in the main text.

Phi-3.5-MoE-Instruct. Tables 8 and 10 report the layer-wise expert overlap statistics for Phi-3.5-MoE-Instruct on TyDiQA and MGSM, respectively. Figures 7 and 8 provide the corresponding global-level routing overlap analysis, serving as the Phi-side counterparts to Figure 1 (a) in the main text.

Despite the substantial architectural differences from Qwen3-30B-A3B (32 vs. 48 layers, top-2 vs. top-8 routing, 16 vs. 128 experts per layer), both the *routing isolation* and *layerwise divergence* phenomena observed in the main text hold consistently: low-resource languages activate a narrow, repetitive set of experts with little overlap at the global level, while the layer-wise similarity follows the same three-phase hierarchy—shallow lay-

Language	Shal.	Mid.	Deep	Avg.
Bengali (BN)	0.12	0.22	0.05	0.13
Swahili (SW)	0.04	0.19	0.04	0.09
Thai (TH)	0.25	0.33	0.12	0.23
Chinese (ZH)	0.36	0.43	0.11	0.30
Japanese (JA)	0.37	0.37	0.10	0.28
Russian (RU)	0.41	0.43	0.17	0.34
German (DE)	0.39	0.46	0.22	0.36
French (FR)	0.46	0.49	0.24	0.40
Spanish (ES)	0.52	0.51	0.25	0.43
Mean	**0.32**	**0.38**	**0.14**	**0.28**

Table 9: Layer-wise expert overlap with English on MGSM (Qwen3-30B-A3B).

Language	Shal.	Mid.	Deep	Avg.
Swahili (SW)	0.10	0.04	0.00	0.04
Thai (TH)	0.05	0.12	0.00	0.06
Bengali (BN)	0.10	0.09	0.06	0.08
Japanese (JA)	0.19	0.30	0.22	0.24
Russian (RU)	0.29	0.40	0.11	0.27
German (DE)	0.24	0.49	0.11	0.28
French (FR)	0.33	0.49	0.06	0.29
Chinese (ZH)	0.05	0.51	0.39	0.32
Spanish (ES)	0.38	0.46	0.11	0.32
Mean	**0.19**	**0.32**	**0.12**	**0.21**

Table 10: Layer-wise expert overlap with English on MGSM (Phi-3.5-MoE-Instruct).

ers showing moderate overlap, middle layers peaking, and deep layers dropping to near zero. These results confirm that the two core phenomena identified in our analysis are robust across model families and evaluation tasks, rather than artifacts of a specific architecture or benchmark.

G RISE Selected Expert Visualization

Figures 9 and 10 visualize the layer-wise distribution of training experts selected by RISE for Bengali on Qwen3-30B-A3B and Phi-3.5-MoE-Instruct, respectively. Each cell indicates whether a given expert in a given layer was selected for training. The heatmaps reveal that RISE concentrates its budget in the middle and deep layers—precisely the layers where Bengali routing diverges most from English (cf. Tables 9–8). This alignment confirms that the composite score in RISE successfully identifies language-specific experts rather than selecting high-activation experts indiscriminately.

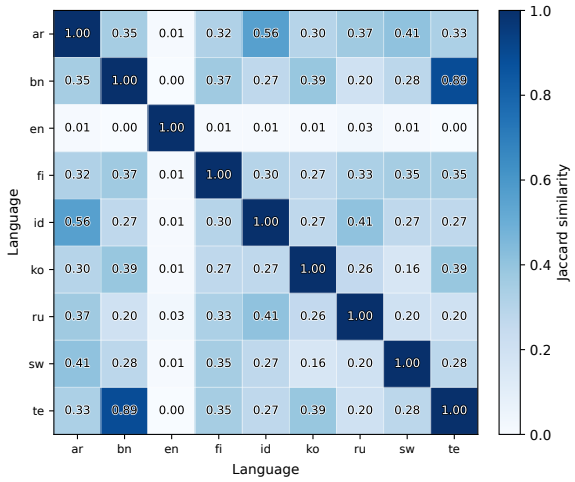


Figure 6: Global-level expert activation overlap of Qwen3-30B-A3B across languages in TyDiQA.

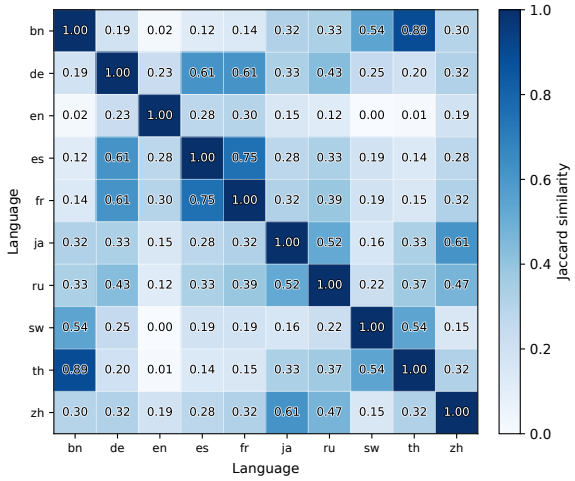


Figure 7: Global-level expert activation overlap of Phi-3.5-MoE across languages in MGSM.

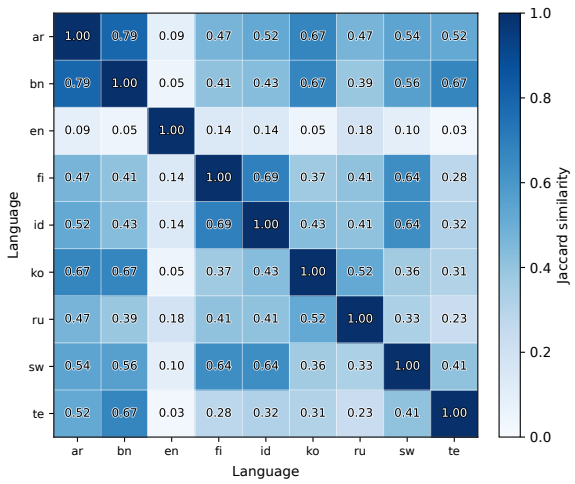


Figure 8: Global-level expert activation overlap of Phi-3.5-MoE across languages in TyDiQA.

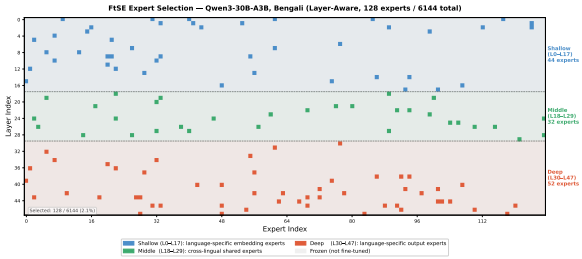


Figure 9: Layer-wise expert selection heatmap for Bengali training on **Qwen3-30B-A3B**. Each column corresponds to a transformer layer and each row to an expert index. Highlighted cells denote experts selected by RISE for gradient updates. The selection density is highest in the middle and deep layers, consistent with the routing isolation observed for Bengali in Table 9.

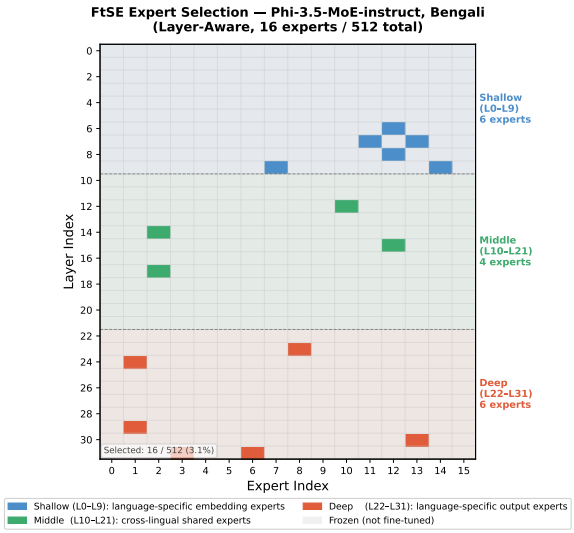


Figure 10: Layer-wise expert selection heatmap for Bengali training on **Phi-3.5-MoE-Instruct**. Despite the architectural differences (32 layers, 16 experts, top-2 routing), RISE again targets experts concentrated in the middle-to-deep range, mirroring the isolation pattern reported in Table 10. The sparser overall selection reflects the smaller expert count per layer relative to Qwen3-30B-A3B.

H Layer Partition and Expert Selection Settings

Table 11 summarizes the layer partition boundaries and expert selection budget used in RISE for both backbone models. The three functional groups—shallow (language-specific encoding), middle (language-agnostic semantic processing), and deep (language-specific generation)—follow the 37.5%/25%/37.5% split derived from our routing divergence analysis in Section 3. The expert budget K is allocated asymmetrically across groups, concentrating the majority of the budget in the middle layers to capture cross-lingual shared computation.

I Why We Do Not Compare with LoRA

We do not compare RISE with LoRA because they operate at orthogonal levels of model adaptation and address fundamentally different questions.

RISE operates at the **routing level**: it identifies *which experts* to update, based on a mechanistic analysis of multilingual routing behavior in MoE models. LoRA operates at the **parameter-update level**: it specifies *how* to update the weight matrices efficiently via low-rank adapters. These are orthogonal design dimensions—RISE concerns the *selection scope* of adaptation, while LoRA concerns the *parameterization* of the update.

The main contribution of this work is the discovery that multilingual MoE models exhibit structured routing isolation, from which language-specific expert subnetworks can be identified in a principled manner. We intentionally train the selected experts with standard full-rank updates to isolate the contribution of expert selection itself, avoiding confounds introduced by rank constraints or adapter initialization.

Importantly, RISE and LoRA are fully compatible rather than mutually exclusive. One may readily apply LoRA, QLoRA, or other PEFT techniques *within* the RISE-selected subnetwork—freezing all non-selected experts and attaching low-rank adapters only to the selected ones. Under this view, RISE answers *which part* of the MoE model to adapt, while LoRA answers *how* to update that part efficiently. Figure 11 illustrates this orthogonal and composable relationship.

For this reason, treating LoRA as a direct competing baseline would conflate two orthogonal design dimensions and blur the central contribution of this paper. We note that prior work has already explored LoRA-style adaptation within MoE training (Deng et al., 2026), further supporting the view that LoRA is a natural *complement* to RISE—applicable

Setting	Qwen3-30B-A3B	Phi-3.5-MoE
Total Layers	48	32
Experts per Layer	128	16
Total Expert Pool	6,144	512
Shallow Layers	0–17 (37.5%)	0–11 (37.5%)
Middle Layers	18–29 (25%)	12–19 (25%)
Deep Layers	30–47 (37.5%)	20–31 (37.5%)
Selected Experts K	128 (2.1%)	16 (3.1%)
Shallow budget	35% (≈44)	12.5% (≈2)
Middle budget	25% (≈32)	68.75% (≈11)
Deep budget	40% (≈52)	18.75% (≈3)
Shallow criterion	Language specificity	Language specificity
Middle criterion	Cross-lingual overlap	Cross-lingual overlap
Deep criterion	Language specificity	Language specificity

Table 11: Layer-aware expert selection settings for Qwen3-30B-A3B and Phi-3.5-MoE-Instruct.

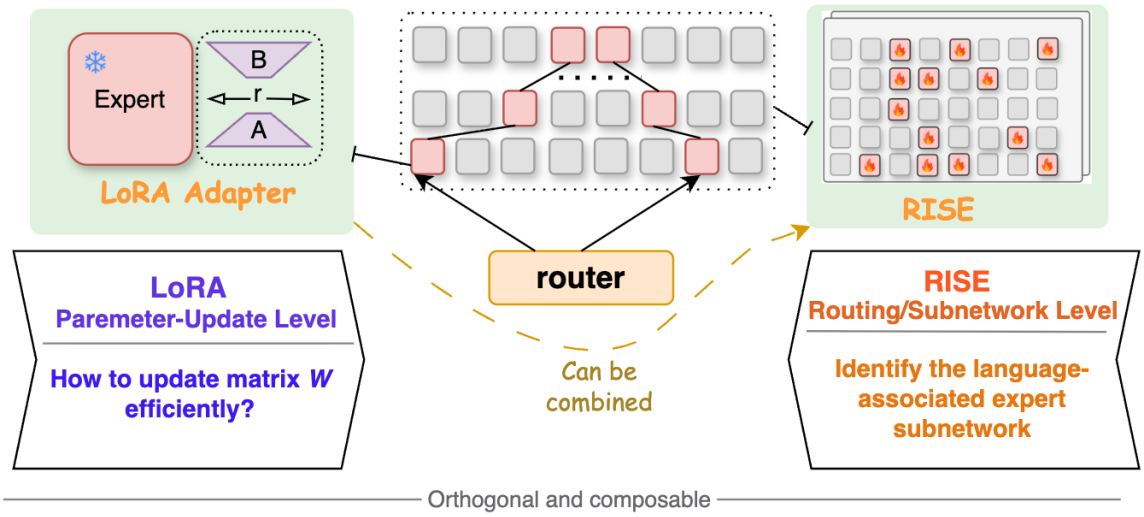


Figure 11: RISE and LoRA are orthogonal and composable: RISE operates at the **routing level** (which experts to adapt), while LoRA operates at the **parameter-update level** (how to update them).

K	Weights	Act.	Grad.	Opt.	Total
8	56.3 GB	7.9 GB	72 MB	432 MB	**64.7 GB**
16	56.3 GB	7.9 GB	144 MB	864 MB	**65.1 GB**
32	56.3 GB	7.9 GB	288 MB	1.7 GB	**66.1 GB**
64	56.3 GB	7.9 GB	576 MB	3.4 GB	**68.1 GB**
128	56.3 GB	7.9 GB	1.1 GB	6.8 GB	**72.0 GB**

Table 12: GPU memory usage during training for different numbers of selected experts K on Qwen3-30B-A3B. Note that these figures represent the *minimum* memory footprint without any training data loaded; actual memory consumption may increase significantly with larger batch sizes.

within the selected subnetwork for additional parameter efficiency—rather than an alternative to it.

J Case Study

Case 1 | Qwen3-30B-A3B / MGSM Bengali — Percentage Misinterpretation

Question:
জোশ একটি বাড়ি ফ্লিপ করার সিদ্ধান্ত নিয়েছেন। তিনি $80,000-এ একটি বাড়ি ক্রয় করলেন এবং $50,000 দিয়ে মেরামত করলেন। এর ফলে বাড়িটির মূল্য 150% বৃদ্ধি পেল। তিনি কত ডলার লাভ করলেন?
Josh bought a house for $80k, repaired it for $50k; value rose by 150%. How much profit?

Vanilla ✗

আসুন প্রশ্নটি ধীরে ধীরে বিশ্লেষণ করি।
প্রথমে মোট খরচ:
বাড়ি ক্রয়: $80,000; মেরামত: $50,000
মোট খরচ $= \$80{,}000 + \$50{,}000 = \$130{,}000$
এরপর নতুন মূল্য:
বাড়িটির মূল্য 150% বৃদ্ধি পেয়েছে।
নতুন মূল্য = পুরানো মূল্য × 150% = $80,000 × **1.5** = **$120,000**
লাভ নির্ণয়:
বিক্রয় মূল্য = $120,000; মোট খরচ = $130,000
লাভ = $120,000 − $130,000 = −$10,000
জোশ কোনো লাভ করেনি, বরং $10,000 লোকসান করলেন।

RISE (Ours) ✓

আসুন প্রশ্নটি ধাপে ধাপে বিশ্লেষণ করি।
প্রথম ধাপ: মোট খরচ
জোশের মোট খরচ:

80,000 + 50,000 = 130,000 ডলার

দ্বিতীয় ধাপ: নতুন মূল্য
বাড়িটির মূল্য 150% বৃদ্ধি পেল, অর্থাৎ:

80,000 × **(1+1.5)** = 80,000 × **2.5** = 200,000 ডলার

তৃতীয় ধাপ: লাভ

200,000 − 130,000 = 70,000 ডলার

উত্তর: **70000**

Gold: $70,000 *Error: "150% increase" interpreted as $\times 1.5$ instead of $\times 2.5$*

Case 2 | Qwen3-30B-A3B / MGSM Bengali — Generation Truncation

Question:
জন এক সপ্তাহে 60 মাইল দৌড়ান। তিনি প্রতি সপ্তাহে 3 দিন দৌড়ান। তিনি প্রথম দিনে 3 ঘণ্টা দৌড়ান এবং অন্য দুদিন তার অর্ধেক দৌড়ান। তিনি কত দ্রুত দৌড়ান?
60 mi/week; Day 1: 3 h, other two days: 1.5 h each. What is his speed?

Vanilla ✗

প্রথম দিন:
সময় = 3 ঘণ্টা; দূরত্ব = 3 ঘণ্টা $\times v$
অন্য দুদিন:
প্রতিদিন সময় = $\frac{3}{2}$ = 1.5 ঘণ্টা
দুদিনের মোট সময় = 1.5 + 1.5 = 3 ঘণ্টা; দূরত্ব = $3v$
মোট দূরত্ব:
$3v + 3v = 6v$; মোট দূরত্ব = 60 মাইল
$6v = 60 \Rightarrow v =$ [*output truncated here*]
Evaluator extracts last number found: ***60***

RISE (Ours) ✓

প্রথম দিনের দৌড়ের সময়: 3 ঘণ্টা
অন্য দুদিনের সময়:
প্রতি দিন 3 ঘণ্টার অর্ধেক = 1.5 ঘণ্টা
দুদিনের মোট সময় = $1.5 \times 2 = 3$ ঘণ্টা
মোট দৌড়ের সময় (3 দিনে):

$3 + 3 = 6$ ঘণ্টা

দ্রুতি:
দ্রুতি $= 60 \div 6 = 10$ মাইল/ঘণ্টা

উত্তর: **10**

Gold: 10 mph *Error: generation cut off; evaluator extracts 60 (distance) not 10 (speed)*

Figure 12: Qualitative case study (**Cases 1–2**) comparing **Vanilla** (base model) and **RISE** (ours) on Bengali MGSM. **Red bold** marks incorrect steps; **green bold** marks correct reasoning. Both cases show mathematical reasoning errors in Qwen3-30B-A3B: Case 1 misinterprets "150% increase" as $\times 1.5$ instead of $\times 2.5$; Case 2 suffers premature generation truncation, causing the evaluator to extract the distance (60) rather than the speed (10). See Figure 13 for Cases 3–4.

Case 3 | Phi-3.5-MoE / TyDiQA Bengali — Language Coherence

Passage & Question:

কর্ণ ক্ষিপ্ত হয়ে অর্জুনের সঙ্গে যুদ্ধে লিপ্ত হন। ভীষণ যুদ্ধ লাগে। দুইজন সমান যোদ্ধা হওয়ার কারণে কেউ কাউকে ছাড় দিচ্ছিলেন না। সবাই নিজ নিজ যুদ্ধ ভুলে অর্জুন এবং কর্ণের যুদ্ধ দেখতে লাগলেন। কর্ণের রথের সারথী ছিলেন শল্য, যিনি ছিলেন সম্পর্কে পাণ্ডবদের মামা। তিনি কর্ণকে নানা কথায় কাবু করতে লাগলেন। হঠাৎ কর্ণের রথের চাকা মাটিতে বসে গেল। [...] পরশুরামের অভিশাপে কর্ণ ব্রহ্মাস্ত্রের মন্ত্র ভুলে গেলেন। শেষ পর্যন্ত অর্জুনের হাতে কর্ণের মৃত্যু হল।

প্রশ্ন: পুরাণ মতে কর্ণ কার হাতে নিহত হন?

(Karna angrily engaged in battle with Arjuna...Karna's chariot wheel sank in mud...Karna was finally slain by Arjuna. Q: According to mythology, at whose hand was Karna slain?)

Vanilla ✗

আ yes, পুরাণ মতে কর্ণ কার হাতে নিহত হয়েছে।

("Ah yes, according to mythology Karna was slain at whose hand.")

Echoes the question verbatim; produces no answer.

RISE (Ours) ✓

অর্জুনের হাতে কর্ণ মৃত্যু হয়।

("Karna was killed at the hand of Arjuna.")

Gold: অর্জুনের (*Arjuna's*) *Error: mixed-language non-answer; question echoed instead of answered*

Case 4 | Phi-3.5-MoE / MGSM Bengali — Incoherent Multi-step Reasoning

Question:

জেমস এক সপ্তাহে 3 বার 3টি স্প্রিন্টে দৌড়ানোর সিদ্ধান্ত নিয়েছেন। তিনি প্রতি স্প্রিন্টে 60 মিটার করে দৌড়ান। তিনি এক সপ্তাহে মোট কত মিটার করে দৌড়ান?

James: 3 sessions/week × 3 sprints/session × 60 m/sprint. Total meters per week?

Vanilla ✗

উপরে উল্লিখিত সমস্যাটি হল জেমস এক সপ্তাহে স্প্রিন্টে দৌড়ানের মোট মিটার গুন নির্ণায়ক।

পদক্ষেপ:

Step 1: প্রতি স্প্রিন্টে দৌড়ানের দূরীয়তা = (3 স্প্রিন্টে দৌড়ান = 60 মিটার)

[*misreads 60 m/sprint as: 3 sprints total = 60 m*]

Step 2: জেমস এক সপ্তাহে স্প্রিন্টে দৌড়ানোর সম্পর্কে জানতে চাহিয়েছে যে সপ্তাহে তিনি দৌড়ানো।

Step 3: এক সপ্তাহে সাত দিন আছে।

[*irrelevant; confuses sessions with days-of-week*]

প্রতি স্প্রিন্টে দৌড়ানের দূরীয়তা [***output truncated; no answer***]

RISE (Ours) ✓

৬০ মিটার × ৩ = ১৮০ মিটার (*60 m/sprint × 3 sprints = 180 m per session*)

১৮০ মিটার × ৩ = ৫৪০ মিটার (*180 m × 3 sessions = 540 m total*)

উত্তর: ৫৪০

Gold: 540 m *Error: misreads per-sprint distance; introduces spurious 7 days/week*

Figure 13: Qualitative case study (**Cases 3–4**) comparing **Vanilla** (base model) and **RISE** (ours) on Bengali TyDiQA and MGSM. **Red bold** marks incorrect steps; **green bold** marks correct reasoning. Both cases show incoherent or mixed-language outputs in Phi-3.5-MoE: Case 3 produces a question-echoing non-answer with mixed-language output; Case 4 misreads per-sprint distance and introduces a spurious 7-day/week step, with output truncated before any answer is given. RISE's Bengali-specialized expert activation corrects both failures.