Routing-Aligned Fine-Tuning for Multilingual Downstream Tasks in Mixture-of-Experts Models

Anonymous ACL submission

Abstract

Mixture-of-Experts (MoE) models have emerged as a dominant paradigm for efficient LLM scaling, yet adapting them to non-English downstream tasks remains challenging. Existing fine-tuning approaches treat MoE models as monolithic learners, ignoring the heterogeneous routing structure that develops during pretraining. We validate across multiple MoE models and downstream tasks that middle layers form a language-universal alignment zone where routing divergence strongly predicts per-language task performance gaps. Building on this observation, we propose **RA-MoE** (Routing-Aligned MoE Fine-Tuning), a three-stage framework that categorizes parallel task examples into a four-way taxonomy (*cc*/*ci*/*ic*/*ii*) based on correctness in English and the target language, identifies task-relevant experts in the middle layers, and augments standard SFT with a routing alignment loss that encourages target-language routing on *ci*-type examples to follow the English task-expert activation pattern. Experiments across three MoE models, three tasks, and six target languages demonstrate that RA-MoE consistently outperforms standard SFT and strong baselines including Routing Steering and RISE, with the *ci* proportion of a task-language pair serving as a reliable predictor of alignment benefit. Code is available at `https://anonymous.4open.science/r/ra-moe-CD06/`.

1 Introduction

Mixture-of-Experts (MoE) has emerged as the dominant paradigm for scaling Large Language Models (LLMs) without proportionally increasing per-token computation (Shazeer et al., 2017; Jiang et al., 2024; Liu et al., 2024). Despite their impressive English-centric capabilities, adapting pretrained MoE models to non-English downstream tasks remains challenging. The prevailing approach is supervised fine-tuning (SFT) on target-language task data (Chen et al., 2024; Chai et al., 2025; Zhang et al., 2026), yet these methods are designed for dense LLMs and treat the model as a monolithic learner, ignoring the heterogeneous routing structure that MoEs develop during pretraining.

A key clue lies in the routing dynamics of pretrained MoEs. Recent studies reveal a consistent U-shaped layer-wise pattern in cross-lingual routing divergence: early and late layers are highly language-specific, while middle layers exhibit strong cross-lingual alignment, with language performance strongly correlated with how similarly a language's tokens are routed to English (Bandarkar et al., 2026; Chen et al., 2026; Zheng et al., 2026). This suggests that middle layers already encode language-universal, task-relevant expertise, and that non-English performance degradation stems partly from the failure to engage these experts for non-English inputs.

We ask: *can we leverage this cross-lingual routing structure to improve MoE fine-tuning on non-English downstream tasks?* We propose **RA-MoE** (**R**outing-**A**ligned **MoE** Fine-Tuning), a three-stage framework. First, we run inference on parallel task data $(x_{\text{en}}, x_{\text{tgt}})$ to categorize examples as *cc*/*ci*/*ic*/*ii* (correct/incorrect in English and target language). Second, we profile per-layer routing distributions and identify task experts in the middle layers using English task and general data. Third, we augment standard SFT with a routing alignment loss applied to *ci*-type examples, encouraging their target-language routing to follow the English task-expert activation pattern. The combined objective balances standard cross-entropy supervision with this routing alignment signal, controlled by a scalar weight λ.

Experiments across three MoE models, three downstream tasks, and six target languages show that RA-MoE consistently outperforms standard SFT and strong baselines, with gains most pronounced on task-language pairs with a higher proportion of *ci*-type examples. Our contributions are:

- We validate the U-shaped cross-lingual routing divergence pattern in task-specific settings, showing that middle-layer routing divergence reliably predicts per-language task performance gaps.

- We introduce a four-category data taxonomy (*cc*/*ci*/*ic*/*ii*) for diagnosing cross-lingual performance gaps, together with empirical evidence that the *ci* proportion predicts the benefit from routing alignment (r=0.70, p=0.001).

- We propose a task expert identification procedure that localizes task-relevant experts in middle MoE layers using English data, and a routing alignment fine-tuning objective that closes the cross-lingual expert activation gap.

- We demonstrate consistent gains over standard SFT and strong baselines (Routing Steering, RISE) across three MoE models (OLMoE-1B-7B, Qwen1.5-MoE, DeepSeek-V2-Lite), three tasks (GSM8K, IFEval, MMLU), and six target languages, with task experts shown to transfer across linguistically distant languages without re-running the identification stage.

2 Related Work

Multilingual downstream task fine-tuning. The standard approach translates English task datasets into target languages and applies SFT (Chen et al., 2024; Chai et al., 2025; Zhang et al., 2026), sometimes augmented with cross-lingual representation alignment objectives (Liu and Niehues, 2025). These methods improve non-English performance but are designed for dense LLMs, offering no mechanism to exploit or align the sparse routing structure of MoEs.

MoE-based multilingual language expansion. A distinct body of work uses MoE architectures to extend the multilingual capabilities of dense LLMs, including upcycling dense models into MoEs with language-specific experts (Zhou et al., 2025), layer-wise expert allocation based on cross-lingual similarity (Zhang et al., 2025), and neuron-level expert allocation (Li and Yanaka, 2026). These methods treat MoE as a tool for augmenting dense models with multilingual capacity, whereas we focus on improving the multilingual downstream task performance of models that are already MoE-based.

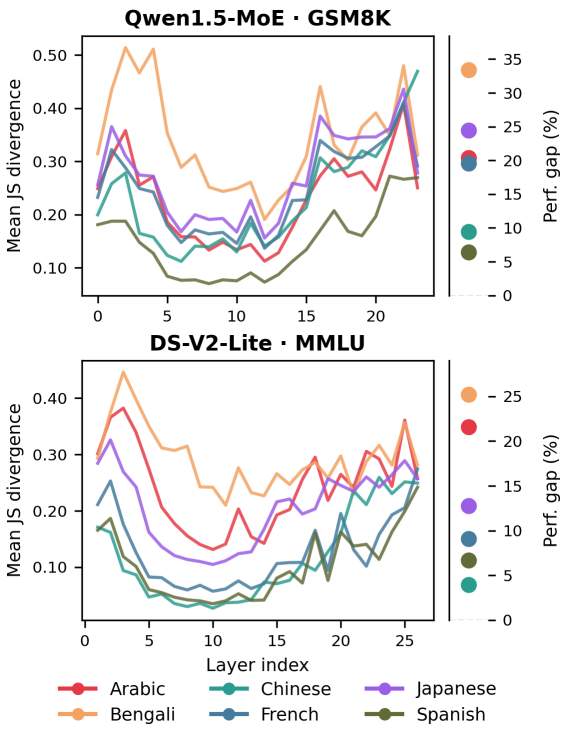

Figure 1: Layer-wise mean JS divergence between English and non-English routing distributions on two MoE models and tasks. Right panels show the task performance gap of each language relative to English.

Multilingual routing analysis in MoEs. Recent studies collectively establish a U-shaped layer-wise pattern in cross-lingual routing divergence, where middle layers exhibit strong cross-lingual alignment strongly correlated with language performance (Bandarkar et al., 2026; Chen et al., 2026). Both works propose inference-time routing-guided steering but do not update model parameters. Zheng et al. (2026) further reveal Language Routing Isolation between high- and low-resource languages, and propose RISE to selectively fine-tune language-specific expert subnetworks.

Routing-aware MoE fine-tuning. RoMA (Li et al., 2025) shows that existing MoE routers are suboptimal on downstream tasks (10–20% accuracy gap relative to oracle routing) and improves generalization by aligning each sample's routing to that of successful neighbors. DR-LoRA (Deng et al., 2026) addresses capacity mismatch by dynamically allocating LoRA ranks based on routing frequency and gradient signals.

While these lines of work collectively highlight the importance of routing structure in multilingual MoE models, none explicitly exploits cross-lingual routing divergence as a fine-tuning signal for non-

English downstream tasks. RA-MoE bridges this gap by identifying task-relevant experts in middle layers and aligning target-language routing toward English patterns on examples where the performance gap is demonstrably language-driven.

3 Method

3.1 Overview

A core premise of RA-MoE is grounded in an empirical observation about routing behavior in pretrained MoEs. Prior work has established a U-shaped layer-wise pattern in cross-lingual routing divergence on general-domain data (Bandarkar et al., 2026; Chen et al., 2026), and we extend this finding to task-specific settings.

Figure 1 confirms that across two representative MoE models and downstream tasks, the Jensen–Shannon divergence between English and non-English routing distributions remains consistently high in early and late layers and substantially reduced in the middle layers, a pattern that holds regardless of task type, language family, or resource level. Critically, languages with larger middle-layer routing divergence consistently exhibit larger task performance gaps (right panels), suggesting that this divergence reflects a language-driven performance gap that is amenable to targeted intervention. This establishes middle layers as a stable, task-relevant cross-lingual alignment zone, and directly motivates the design of RA-MoE.

RA-MoE proceeds in three stages, as illustrated in Figure 2. Stage 1 constructs parallel task data and partitions examples into four categories based on correctness in English and the target language. Stage 2 analyzes the MoE model's routing behavior to identify the middle layers exhibiting cross-lingual alignment and the task-relevant experts within those layers. Stage 3 fine-tunes the MoE model on target-language task data, augmenting the standard cross-entropy loss with a routing alignment loss applied selectively to *ci*-type examples, for which the model answers correctly in English but fails in the target language.

3.2 Stage 1: Parallel Data Construction and Categorization

Parallel data construction. Let $\mathcal{D}_{\text{en}} = \{(x_{\text{en}}^{(i)}, y_{\text{en}}^{(i)})\}_{i=1}^{N}$ denote an English downstream task dataset. We translate each sample into the target language, yielding parallel pairs $\{(x_{\text{en}}^{(i)}, x_{\text{tgt}}^{(i)})\}_{i=1}^{N}$ for routing profiling in Stage 2 and $\mathcal{D}_{\text{tgt}} = \{(x_{\text{tgt}}^{(i)}, y_{\text{tgt}}^{(i)})\}_{i=1}^{N}$ for fine-tuning in Stage 3.

Four-category taxonomy. We run inference with the MoE model $\mathcal{M}$ on both $x_{\text{en}}^{(i)}$ and $x_{\text{tgt}}^{(i)}$, and collect the model's responses $\hat{y}_{\text{en}}^{(i)}$ and $\hat{y}_{\text{tgt}}^{(i)}$. We then partition the parallel dataset into four categories based on correctness: ***cc*** (both correct), ***ci*** (English correct, target incorrect), ***ic*** (English incorrect, target correct), and ***ii*** (both incorrect). As we show in Section 4.4, for tasks where $\mathcal{M}$ achieves strong English performance, *ci* consistently accounts for the largest share of cross-lingual errors, reflecting a language-driven performance gap where the model possesses the requisite task knowledge but fails to apply it in the target language.

3.3 Stage 2: Middle Layer and Task Expert Identification

Routing profiling. To analyze the model's routing behavior, we run teacher-forcing inference on the concatenation of each prompt and its Stage 1 generated response $(x_{\text{en}}^{(i)}, \hat{y}_{\text{en}}^{(i)})$ and $(x_{\text{tgt}}^{(i)}, \hat{y}_{\text{tgt}}^{(i)})$ respectively, and record the routing distributions at each MoE layer. Specifically, at each layer ℓ, we register a forward hook on the router module to capture the post-softmax routing weight vector $\mathbf{p}_t^{(\text{lang},\ell)} \in [0,1]^E$ for each token t, and average over generated token positions to obtain a sequence-level routing distribution:

$$\mathbf{q}^{(\text{lang},\ell,i)} = \frac{1}{|G^{(i)}|} \sum_{t \in G^{(i)}} \mathbf{p}_t^{(\text{lang},\ell,i)}, \qquad (1)$$

where $G^{(i)}$ denotes the set of generated token positions for sample i.

Middle layer identification. To identify which layers are suitable targets for routing alignment, we compute the mean layer-wise routing divergence across all N parallel pairs:

$$\overline{\text{Div}}^{(\ell)} = \frac{1}{N} \sum_{i=1}^{N} D_{\text{JS}}\left(\mathbf{q}^{(\text{en},\ell,i)} \,\middle\|\, \mathbf{q}^{(\text{tgt},\ell,i)}\right), \qquad (2)$$

where D_{JS} denotes the entropy-normalized Jensen–Shannon divergence. As shown in Figure 1, this divergence exhibits a characteristic U-shaped profile across layers, with substantially lower divergence in the middle layers across all three models and tasks. We define the middle layer range $\mathcal{L}_{\text{mid}} = \{m, \ldots, n\}$ as the longest contiguous segment of layers whose mean divergence falls below

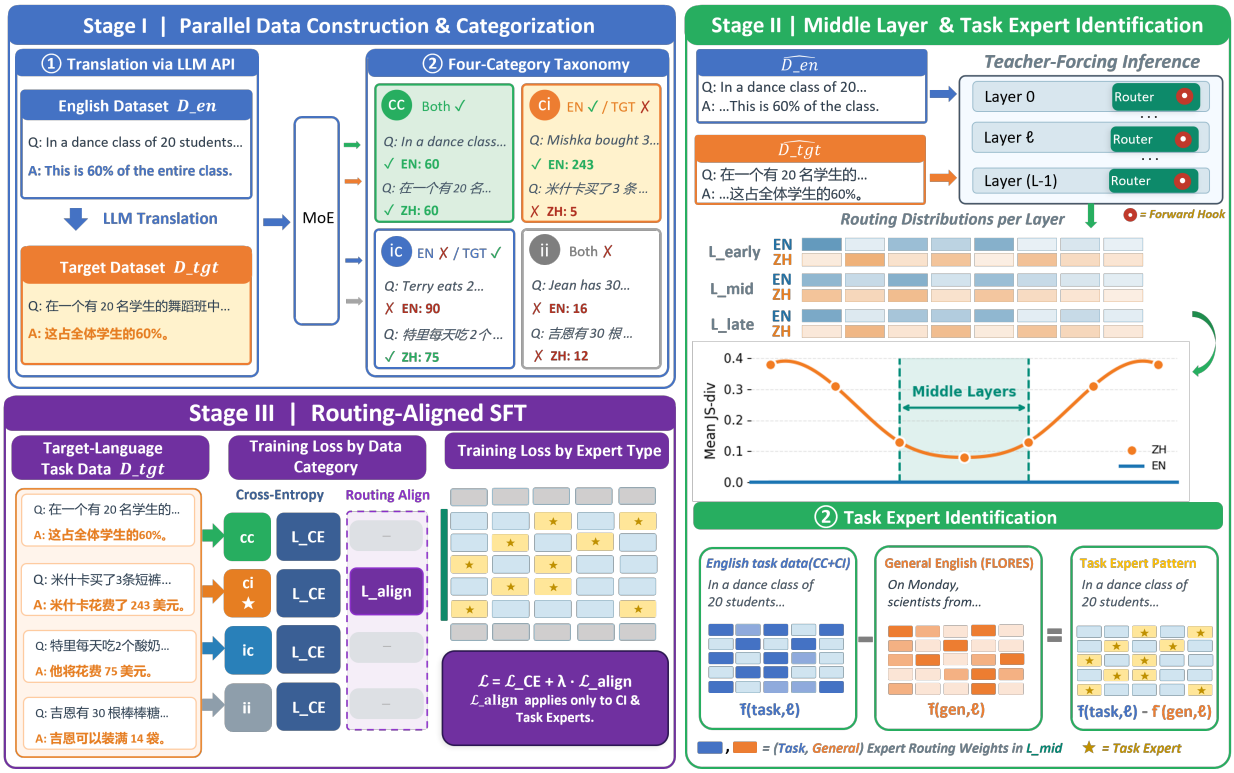

Figure 2: Overview of the RA-MoE framework.

the median of the per-layer divergence distribution, a threshold that adapts automatically to each model's divergence profile and consistently identifies approximately the middle third of transformer layers across all three models in our experiments.

Task expert identification. We identify a set of *task experts* in each middle layer $\ell \in \mathcal{L}_{\text{mid}}$—the experts that $\mathcal{M}$ preferentially activates when solving the downstream task in English. To do so, we contrast two data sources: the English task data $\mathcal{D}_{\text{en}}^{\text{task}}$ (restricted to correctly answered examples, i.e., *cc* and *ci* examples from Stage 1) and a general English corpus $\mathcal{D}_{\text{en}}^{\text{gen}}$ (FLORES-200 English, Goyal et al. 2022). For each source, we compute the mean routing weight per expert:

$$\bar{f}_e^{(\text{src},\ell)} = \frac{1}{|\mathcal{D}_{\text{src}}|} \sum_{i \in \mathcal{D}_{\text{src}}} q_e^{(\text{src},\ell,i)}, \tag{3}$$

and define the task-specificity score as:

$$\Delta_e^{(\ell)} = \bar{f}_e^{(\text{task},\ell)} - \bar{f}_e^{(\text{gen},\ell)}. \tag{4}$$

For each middle layer, we select the top-K experts by $\Delta_e^{(\ell)}$ among those with $\Delta_e^{(\ell)} > 0$ as the task expert set $\mathcal{E}_\ell^{\text{task}}$. Experts with $\Delta_e^{(\ell)} \leq 0$ are excluded even if they appear in the top-K, as a non-positive score indicates no task-specific preference. In our experiments we use $K = 8$; sensitivity to this choice is analyzed in Section 4.3.

Finally, for each *ci* example i, we store the English routing distributions $\{\mathbf{q}^{(\text{en},\ell,i)}\}_{\ell \in \mathcal{L}_{\text{mid}}}$ as reference signals for Stage 3.

3.4 Stage 3: Routing-Aligned SFT

Standard cross-entropy loss. We fine-tune $\mathcal{M}$ on $\mathcal{D}_{\text{tgt}}$ with the standard next-token prediction objective:

$$\mathcal{L}_{\text{CE}} = -\sum_i \sum_t \log p_\theta\left(y_t^{(i)} \mid x_{\text{tgt}}^{(i)}, y_{<t}^{(i)}\right). \tag{5}$$

Routing alignment loss. For *ci*-type examples, we additionally encourage the model's middle-layer routing on the target-language input to approach the English reference routing stored in Stage 2. Concretely, for each *ci* example i and each middle layer $\ell \in \mathcal{L}_{\text{mid}}$, we restrict both routing distributions to the task expert set $\mathcal{E}_\ell^{\text{task}}$ and renormalize for each $e \in \mathcal{E}_\ell^{\text{task}}$:

$$\tilde{q}_e^{(\text{lang},\ell,i)} = \frac{q_e^{(\text{lang},\ell,i)}}{\sum_{e' \in \mathcal{E}_\ell^{\text{task}}} q_{e'}^{(\text{lang},\ell,i)}}. \tag{6}$$

and compute the KL divergence from the fixed English reference to the current target-language

routing ($\tilde{\mathbf{q}}^{(\text{lang},\ell,i)} = (\tilde{q}_e^{(\text{lang},\ell,i)})_{e\in\mathcal{E}_\ell^{\text{task}}}$):

$$\mathcal{L}_{\text{align}} = \sum_{i\in\mathcal{D}_{\text{ci}}} \sum_{\ell\in\mathcal{L}_{\text{mid}}} D_{\text{KL}}\left(\tilde{\mathbf{q}}^{(\text{en},\ell,i)} \,\|\, \tilde{\mathbf{q}}^{(\text{tgt},\ell,i)}\right). \tag{7}$$

Using the English distribution as the reference ensures that the gradient signal encourages the target-language routing to move toward the English pattern, which is the direction of task-relevant knowledge transfer. Restricting to $\mathcal{E}_\ell^{\text{task}}$ focuses the alignment signal on task-relevant experts within the middle layers, avoiding interference with other experts that are not preferentially activated for the downstream task.

Combined objective. The final training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{align}}, \tag{8}$$

where $\lambda > 0$ controls the strength of the alignment signal. The alignment loss is computed only for *ci*-type examples within each training batch; *cc*, *ic*, and *ii* examples contribute only to $\mathcal{L}_{\text{CE}}$. This selective application ensures that the routing alignment signal is grounded in cases where the English routing pattern is a valid and reliable reference, specifically examples where the model already solves the task correctly in English.

4 Experiments

4.1 Experimental Setup

Models. We evaluate RA-MoE on three MoE models of varying architectures and scales: **OLMoE-1B-7B-0924-SFT** (Muennighoff et al., 2025, hereafter OLMoE), **Qwen1.5-MoE-A2.7B** (Team, 2024, Qwen1.5-MoE), and **DeepSeek-V2-Lite-Chat** (DeepSeek-AI et al., 2024, DS-V2-Lite). All models share identical training schedules, optimization settings, and data orders.

Tasks and languages. We fine-tune the selected models on three downstream tasks using LoRA, each paired with standardized benchmarks: **Mathematical Reasoning** (GSM8K; Cobbe et al. 2021), **Instruction Following** (IFEval; Zhou et al. 2023), and **Knowledge Understanding** (MMLU; Hendrycks et al. 2021). For each task, we translate both the English training and evaluation data into six target languages: **Arabic** (ar), **Bengali** (bn), **Chinese** (zh), **French** (fr), **Japanese** (ja), and **Spanish** (es), covering a range of language families and resource levels. Data construction and translation details are provided in Appendix A.

Baselines. We compare RA-MoE against the following baselines:

- **Zero-shot**: The MoE model evaluated directly on target-language inputs without any task-specific fine-tuning.
- **SFT**: Standard supervised fine-tuning on target-language task data with cross-entropy loss only.
- **Routing Steering** (Bandarkar et al., 2026): Inference-time intervention that promotes English-aligned expert activation in middle layers, without updating model parameters.
- **RISE** (Zheng et al., 2026): Fine-tuning that selectively updates language-specific expert subnetworks identified by routing analysis, without a cross-lingual alignment signal.

Implementation details and hyperparameter settings are provided in Appendix B.

4.2 Main Results

Table 1 presents the main results of RA-MoE and all baselines across three models, three tasks, and six target languages. We report mean accuracy with standard deviations over 5 random seeds.

Several observations stand out. First, RA-MoE consistently outperforms standard SFT across all three models and tasks, demonstrating that routing alignment provides a reliable improvement over naive target-language fine-tuning. Second, RA-MoE substantially outperforms Routing Steering by leveraging routing alignment as a persistent training signal rather than a one-time inference-time patch, while RS fails to match SFT in most settings, as inference-time interventions applied to the base model without any task-specific fine-tuning yield only marginal gains over zero-shot performance. Third, RA-MoE outperforms RISE in the majority of settings, indicating that grounding the alignment signal in the English routing pattern on *ci* examples is more effective than selecting which experts to update without specifying how they should route. Finally, gains are most pronounced on tasks and language pairs with a higher proportion of *ci*-type examples (see Section 4.4.1), consistent with our hypothesis that the *ci* proportion reflects the degree to which the performance gap is language-driven rather than knowledge-driven.

Model	Method	GSM8K						IFEval						MMLU						Avg.
		ar	bn	zh	fr	ja	es	ar	bn	zh	fr	ja	es	ar	bn	zh	fr	ja	es	
OLMoE	0-shot	3.8 ±0.1	1.9 ±0.2	17.7 ±0.4	20.7 ±0.4	7.2 ±0.3	23.0 ±0.2	13.7 ±0.3	11.7 ±0.4	13.8 ±0.3	23.7 ±0.3	17.5 ±0.5	23.1 ±0.5	28.3 ±0.1	25.5 ±0.1	31.7 ±0.1	37.6 ±0.1	29.6 ±0.1	36.9 ±0.1	20.4 ±0.3
	SFT	9.4 ±0.2	**2.2 ±0.1**	21.3 ±0.3	25.7 ±0.3	13.8 ±0.3	28.3 ±0.2	18.1 ±0.4	14.1 ±0.4	17.8 ±0.3	27.8 ±0.4	21.3 ±0.3	27.5 ±0.3	28.4 ±0.2	25.6 ±0.1	32.1 ±0.2	38.0 ±0.1	29.7 ±0.1	37.8 ±0.1	23.3 ±0.2
	RS	5.1 ±0.1	2.0 ±0.1	18.6 ±0.3	21.3 ±0.3	6.8 ±0.3	23.6 ±0.2	13.3 ±0.3	12.4 ±0.4	14.4 ±0.3	24.5 ±0.3	17.8 ±0.4	24.5 ±0.4	27.5 ±0.1	25.0 ±0.1	31.3 ±0.1	37.1 ±0.1	29.5 ±0.1	37.3 ±0.1	20.7 ±0.2
	RISE	9.7 ±0.2	2.0 ±0.1	21.7 ±0.3	25.7 ±0.3	14.2 ±0.3	28.1 ±0.2	18.4 ±0.4	14.5 ±0.3	17.5 ±0.3	28.2 ±0.4	21.6 ±0.3	27.9 ±0.3	**29.1 ±0.1**	25.8 ±0.1	**32.9 ±0.1**	38.2 ±0.1	29.9 ±0.1	38.1 ±0.1	23.5 ±0.2
	RA-MoE	**10.4 ±0.2**	**2.2 ±0.2**	**22.5 ±0.3**	**26.5 ±0.3**	**15.2 ±0.3**	**29.2 ±0.2**	**19.2 ±0.4**	**15.5 ±0.3**	**18.9 ±0.3**	**28.9 ±0.4**	**22.5 ±0.4**	**28.7 ±0.3**	28.9 ±0.1	**26.0 ±0.1**	32.7 ±0.1	**38.6 ±0.1**	**30.3 ±0.1**	**38.6 ±0.1**	**24.2 ±0.2**
Qwen1.5	0-shot	29.3 ±0.5	3.2 ±0.2	28.8 ±0.3	41.8 ±0.5	9.4 ±0.6	44.4 ±0.4	18.8 ±0.4	18.9 ±0.4	20.5 ±0.3	28.9 ±0.4	22.9 ±0.5	28.4 ±0.3	35.8 ±0.1	24.3 ±0.1	46.7 ±0.1	45.2 ±0.1	37.9 ±0.2	45.4 ±0.1	29.5 ±0.3
	SFT	48.8 ±0.4	29.5 ±0.6	56.7 ±0.3	57.2 ±0.6	42.3 ±0.4	58.5 ±0.6	21.4 ±0.5	20.5 ±0.4	22.3 ±0.5	28.9 ±0.3	25.8 ±0.4	29.6 ±0.4	40.0 ±0.1	28.3 ±0.1	52.8 ±0.2	50.6 ±0.1	42.7 ±0.1	51.2 ±0.1	39.3 ±0.3
	RS	29.5 ±0.5	4.1 ±0.2	29.4 ±0.3	41.3 ±0.5	9.9 ±0.5	45.1 ±0.4	19.1 ±0.4	19.9 ±0.4	20.0 ±0.3	29.3 ±0.4	23.5 ±0.5	28.9 ±0.3	36.4 ±0.1	24.9 ±0.1	46.0 ±0.1	45.8 ±0.1	38.3 ±0.1	45.1 ±0.1	29.8 ±0.3
	RISE	50.1 ±0.3	30.6 ±0.6	57.5 ±0.3	57.7 ±0.6	43.4 ±0.4	59.0 ±0.5	21.9 ±0.5	21.1 ±0.4	22.7 ±0.5	29.4 ±0.4	26.2 ±0.4	**30.7 ±0.4**	40.3 ±0.1	28.6 ±0.1	53.2 ±0.1	51.0 ±0.1	43.1 ±0.1	**52.3 ±0.1**	39.9 ±0.3
	RA-MoE	**51.7 ±0.3**	**32.5 ±0.6**	**59.2 ±0.3**	**59.1 ±0.3**	**45.8 ±0.4**	**60.3 ±0.4**	**23.0 ±0.4**	**21.7 ±0.3**	**23.8 ±0.4**	**29.8 ±0.4**	**27.1 ±0.4**	30.6 ±0.4	**41.2 ±0.1**	**29.4 ±0.2**	**53.8 ±0.1**	**51.6 ±0.1**	**43.8 ±0.1**	52.1 ±0.1	**40.9 ±0.3**
DSV2	0-shot	26.0 ±0.6	10.2 ±0.5	63.7 ±0.3	58.7 ±0.5	43.7 ±0.6	61.4 ±0.5	14.4 ±0.5	11.7 ±0.4	15.1 ±0.2	27.4 ±0.5	19.5 ±0.5	26.5 ±0.6	32.1 ±0.2	28.8 ±0.1	49.9 ±0.1	44.7 ±0.1	40.8 ±0.1	47.4 ±0.1	34.6 ±0.4
	SFT	37.6 ±0.4	19.4 ±0.4	64.2 ±0.2	58.6 ±0.4	50.0 ±0.4	63.4 ±0.1	18.2 ±0.4	15.3 ±0.3	24.6 ±0.4	30.5 ±0.3	26.4 ±0.3	30.0 ±0.2	32.3 ±0.1	29.2 ±0.2	50.0 ±0.1	45.0 ±0.1	42.2 ±0.2	48.2 ±0.1	38.1 ±0.3
	RS	26.4 ±0.5	11.6 ±0.4	63.1 ±0.3	59.3 ±0.5	44.2 ±0.5	61.7 ±0.4	14.7 ±0.5	13.0 ±0.4	14.9 ±0.2	28.1 ±0.5	19.4 ±0.5	26.9 ±0.5	32.5 ±0.2	28.9 ±0.1	49.3 ±0.1	44.4 ±0.1	41.1 ±0.1	47.9 ±0.1	34.9 ±0.3
	RISE	37.9 ±0.3	19.9 ±0.4	64.4 ±0.2	58.9 ±0.4	50.3 ±0.4	63.6 ±0.2	18.5 ±0.4	15.6 ±0.3	25.0 ±0.4	30.9 ±0.4	26.7 ±0.4	30.2 ±0.2	**32.9 ±0.1**	29.3 ±0.1	50.1 ±0.1	**45.2 ±0.1**	42.5 ±0.1	48.3 ±0.1	38.3 ±0.3
	RA-MoE	**39.0 ±0.3**	**20.7 ±0.3**	**65.2 ±0.2**	**59.9 ±0.5**	**51.6 ±0.4**	**64.3 ±0.2**	**19.6 ±0.4**	**16.5 ±0.3**	**25.8 ±0.4**	**31.6 ±0.4**	**27.3 ±0.3**	**31.4 ±0.4**	32.8 ±0.1	**29.7 ±0.1**	**50.6 ±0.1**	45.0 ±0.1	**43.2 ±0.1**	**48.8 ±0.1**	**39.1 ±0.3**

Table 1: Main results (%) across three tasks, three models, and six target languages. OLMoE, Qwen1.5, and DSV2 refer to OLMoE-1B-7B-0924-SFT, Qwen1.5-MoE-A2.7B, and DeepSeek-V2-Lite-Chat, respectively. RS denotes Routing Steering (Bandarkar et al., 2026). The **Avg.** column averages over all 18 task-language combinations. Best result per column is **bolded**; second best is underlined.

Method	bn	ar	es
w/o $\mathcal{L}_{\text{align}}$	29.5 (0.6)	48.8 (0.4)	58.5 (0.6)
w/o task experts	31.2 (0.5)	50.1 (0.4)	59.4 (0.5)
w/o *ci* filtering	31.6 (0.6)	50.6 (0.3)	59.7 (0.4)
w/o mid. layers	29.8 (0.7)	49.2 (0.5)	58.8 (0.5)
RA-MoE	**32.5 (0.6)**	**51.7 (0.3)**	**60.3 (0.4)**

Table 2: Ablation study on Qwen1.5-MoE (GSM8K). w/o $\mathcal{L}_{\text{align}}$ is equivalent to standard SFT. Best result per column is **bolded**.

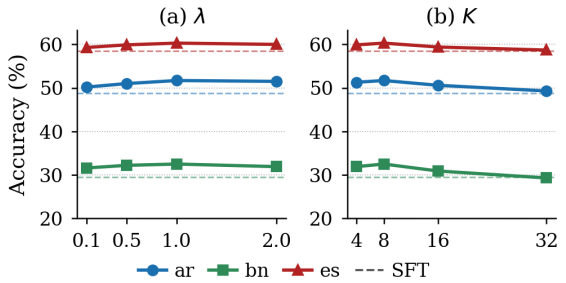

Figure 3: Sensitivity analysis on Qwen1.5-MoE (GSM8K). Solid lines show RA-MoE; dashed lines indicate SFT baseline. Left: varying λ with K=8. Right: varying K with λ=1.0.

4.3 Ablation Studies

To assess the contribution of each component of RA-MoE, we conduct ablation experiments on Qwen1.5-MoE using GSM8K across Bengali, Arabic, and Spanish as representatives of low-, medium-, and high-resource languages. Table 2 reports the following variants:

- **w/o $\mathcal{L}_{\text{align}}$**: Remove the routing alignment loss entirely, reducing RA-MoE to standard SFT.
- **w/o task experts**: Apply $\mathcal{L}_{\text{align}}$ over the full routing distribution (all experts in $\mathcal{L}_{\text{mid}}$) rather than restricting to $\mathcal{E}_{\ell}^{\text{task}}$.
- **w/o *ci* filtering**: Apply $\mathcal{L}_{\text{align}}$ to all training examples rather than *ci*-type only.
- **w/o middle layers**: Apply $\mathcal{L}_{\text{align}}$ across all MoE layers rather than restricting to $\mathcal{L}_{\text{mid}}$.
- **RA-MoE**: Our complete method.

Removing the alignment loss entirely (i.e., reverting to SFT) causes the largest single drop, confirming that routing alignment is the primary driver of improvement. Extending alignment beyond middle layers also leads to a substantial degradation, consistent with the finding that early and late layers are language-specific and should not be constrained toward English routing. Replacing task experts with the full routing distribution reduces performance, indicating that focusing on task-relevant experts provides a cleaner and more informative alignment target. Finally, removing *ci* filtering also hurts, as including *cc*, *ic*, and *ii* examples introduces alignment targets that are either redundant (*cc*) or unreliable (*ic*, *ii*).

Sensitivity to λ and K. Figure 3 reports performance as a function of the alignment weight $\lambda \in \{0.1, 0.5, 1.0, 2.0\}$ and the number of task experts per layer $K \in \{4, 8, 16, 32\}$ on Qwen1.5-MoE (GSM8K). RA-MoE is robust across a wide range of both hyperparameters, with λ=1.0 and K=8 performing consistently well across all three languages. Too small a λ weakens the routing alignment signal, while too large a λ interferes with the cross-entropy learning signal. Similarly, too small

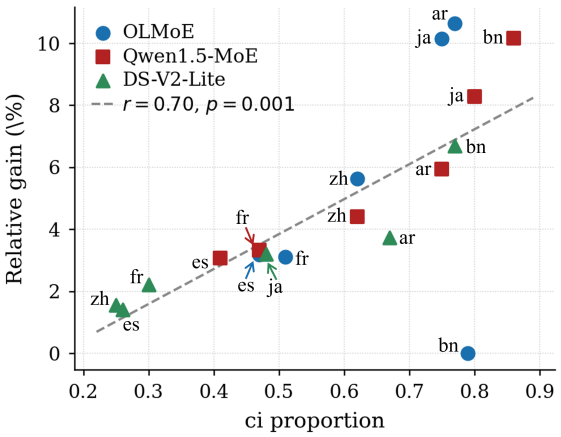

Figure 4: Relative gain of RA-MoE over SFT vs. ci proportion across six target languages and three models (GSM8K). The dashed line shows the linear fit.

		GSM8K		IFEval		MMLU	
Model	**Method**	**ar**	**bn**	**ar**	**bn**	**ar**	**bn**
OLMoE	SFT	9.4 ±0.3	2.2 ±0.1	18.1 ±0.4	14.1 ±0.3	28.4 ±0.1	25.6 ±0.1
	RA-MoE	10.4 ±0.2	2.2 ±0.2	19.2 ±0.4	15.5 ±0.3	28.9 ±0.1	26.0 ±0.1
	RA-MoE†	10.0 ±0.3	2.1 ±0.1	18.9 ±0.4	15.2 ±0.3	28.7 ±0.2	25.8 ±0.1
Qwen1.5	SFT	48.8 ±0.5	29.5 ±0.4	21.4 ±0.3	20.5 ±0.4	40.0 ±0.1	28.3 ±0.1
	RA-MoE	51.7 ±0.3	32.5 ±0.6	23.0 ±0.4	21.7 ±0.3	41.2 ±0.1	29.4 ±0.2
	RA-MoE†	50.7 ±0.5	31.6 ±0.4	22.5 ±0.4	21.2 ±0.3	40.8 ±0.1	29.0 ±0.1
DSV2	SFT	37.6 ±0.4	19.4 ±0.3	18.2 ±0.3	15.3 ±0.4	32.3 ±0.1	29.2 ±0.1
	RA-MoE	39.0 ±0.3	20.7 ±0.3	19.6 ±0.4	16.5 ±0.3	32.8 ±0.1	29.7 ±0.1
	RA-MoE†	38.5 ±0.4	20.2 ±0.3	18.9 ±0.4	16.0 ±0.3	32.6 ±0.1	29.5 ±0.1

Table 3: Cross-language generalization results (%) with standard deviations. RA-MoE† uses task experts identified on zh and applies them directly to ar and bn fine-tuning without re-identifying task experts.

a K fails to cover sufficient task-relevant experts, while too large a K introduces task-irrelevant experts that dilute the alignment target.

4.4 Analysis

4.4.1 Effect of *ci* Proportion

We examine whether the proportion of *ci* examples in a task-language pair predicts the gain of RA-MoE over standard SFT. Figure 4 plots the relative improvement of RA-MoE over SFT against the *ci* proportion across six target languages and three models on GSM8K. A clear positive correlation emerges ($r = 0.70, p = 0.001$): pairs with a higher *ci* proportion benefit more from routing alignment. This is consistent with our design: when the performance gap is primarily language-driven (large *ci*), the English routing pattern provides a strong and reliable supervisory signal; when the gap is knowledge-driven (large *ii*), routing alignment has less to offer. A notable exception is OLMoE on Bengali, which exhibits a high *ci* proportion yet yields no improvement: we attribute this to Bengali being severely under-represented in OLMoE's pretraining corpus, such that the model lacks the Bengali language capacity necessary to benefit from routing alignment. The *ci* proportion thus serves as a practical predictor of how much a given task-language pair stands to benefit from RA-MoE, provided the model has sufficient pretraining coverage of the target language.

4.4.2 Routing Divergence Before and After Fine-Tuning

Figure 5 visualizes the layer-wise mean routing divergence between English and target-language inputs before and after fine-tuning Qwen1.5-MoE on GSM8K. Before fine-tuning, all three target languages (ar, bn and zh) exhibit substantial middle-layer divergence. After standard SFT, divergence decreases marginally, indicating that cross-entropy training alone does little to align routing patterns. In contrast, RA-MoE substantially reduces middle-layer routing divergence across all three languages, bringing target-language routing closer to the English reference. This confirms that the routing alignment loss achieves its intended effect at the routing level, and that the performance gains are accompanied by the expected mechanistic change.

4.4.3 Task Expert Activation During Training

Figure 6 tracks two complementary signals throughout training on two representative model–language pairs (DS-V2-Lite on Arabic and Qwen1.5-MoE on Bengali): the eval CE loss on target-language test data and the task expert selection rate on target-language inputs.

Under standard SFT, the task expert selection rate increases gradually as the model adapts to the target language, yet this improvement is driven purely by cross-entropy supervision with no explicit routing incentive. Under RA-MoE, the selection rate rises more steeply and converges to a higher plateau, confirming that the routing alignment loss successfully steers the model toward task-relevant experts identified in Stage 2. The benefit is also reflected in eval CE loss: RA-MoE consistently achieves lower CE loss than SFT from the mid-training stage onward, suggesting that better engagement of task experts translates into improved target-language generalization.

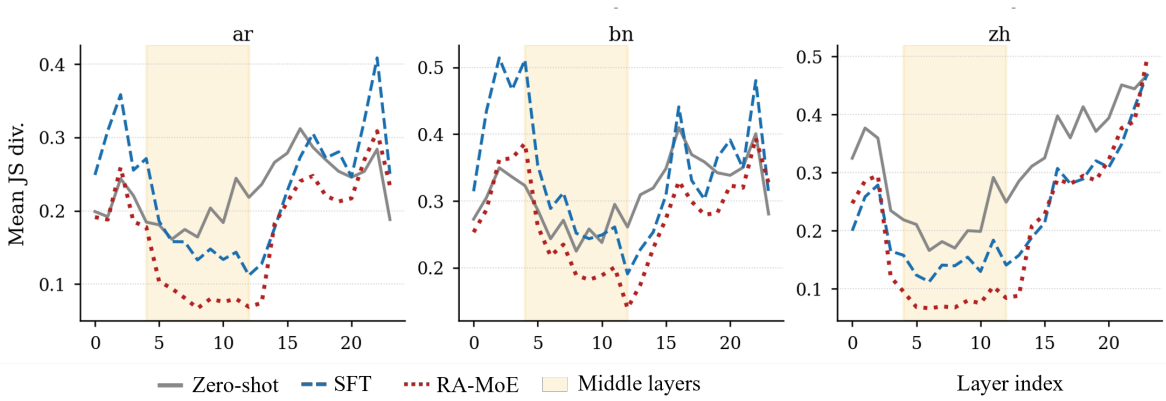

Figure 5: Layer-wise mean routing divergence between English and target-language inputs on Qwen1.5-MoE (GSM8K). The shaded region indicates the middle layers (layers 4–12) identified by RA-MoE.

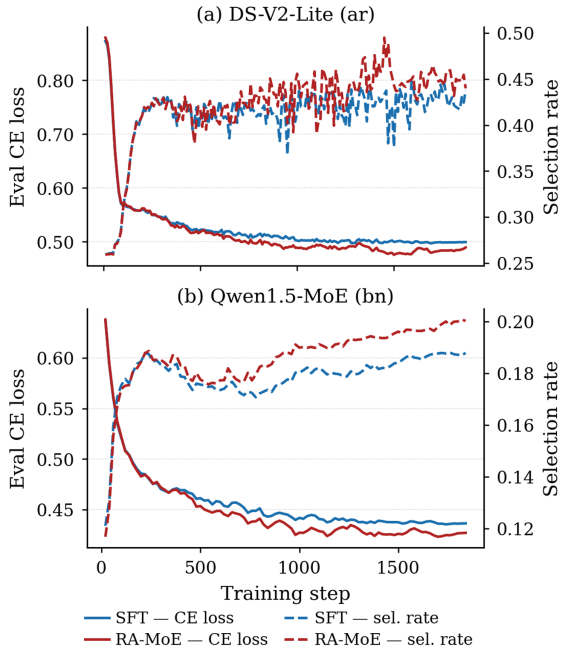

Figure 6: Eval CE loss (solid lines, left axis) and task expert selection rate (dashed, right) throughout training for SFT and RA-MoE on two model–language pairs.

4.4.4 Generalization Across Languages

To assess whether task experts identified in Stage 2 are truly language-universal, we conduct a cross-language generalization experiment: we identify task experts using routing profiling on one source language (Chinese), and apply RA-MoE to fine-tune on a different target language (Arabic or Bengali) using the same task expert set, without re-running the task expert identification step. Table 3 compares this transfer setting against the standard RA-MoE setup (where task experts are identified for each target language independently). Performance under the transfer setting is close to that of the full pipeline, with only a modest drop, suggesting that middle-layer task experts are largely language-agnostic and transfer well across linguistically distant language pairs. This is consistent with our finding that middle layers serve as a language-universal alignment zone, where task-relevant experts remain stable across target languages. This property reduces the computational cost of deploying RA-MoE to new languages, as task expert identification need not be rerun per language.

5 Conclusion

We presented RA-MoE, a three-stage routing-aligned fine-tuning framework that improves non-English downstream task performance in MoE models by explicitly aligning target-language routing toward English task-expert activation patterns in the middle layers. Our work is grounded in a task-specific validation of the U-shaped cross-lingual routing divergence pattern, showing that middle-layer routing divergence reliably predicts per-language task performance gaps across models and tasks. Experiments across three MoE models, three tasks, and six target languages show that RA-MoE consistently outperforms standard SFT and strong baselines including Routing Steering and RISE, with gains most pronounced on task-language pairs with a higher *ci* proportion. Further analysis confirms that the *ci* proportion of a task-language pair reliably predicts the benefit from routing alignment, and that the identified task experts transfer well across linguistically distant languages without re-running the expert identification stage, suggesting that middle-layer task expertise in MoEs is largely language-agnostic.

Limitations

While RA-MoE demonstrates consistent improvements across a range of MoE models, tasks, and target languages, several limitations remain. First, our method presupposes that the base MoE model achieves sufficiently strong English task performance: when English capability is weak, the English routing patterns carry limited task-relevant information and therefore cannot serve as a reliable supervisory signal for target-language inputs, which undermines the effectiveness of routing alignment. Second, even when English routing provides a strong signal, RA-MoE offers limited gains for languages that are severely under-represented in the model's pretraining corpus. In such cases, the performance bottleneck lies not in routing misalignment but in the absence of adequate language-specific capacity, so steering the model toward task-relevant experts cannot compensate for the lack of foundational multilingual competence. Third, our experiments are conducted exclusively on text-only MoE models; whether the U-shaped cross-lingual routing divergence pattern and the corresponding alignment strategy generalize to multimodal large language models remains an open question that we leave for future work.

References

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*.

Linzheng Chai, Jian Yang, Tao Sun, Hongcheng Guo, Jiaheng Liu, Bing Wang, Xinnian Liang, Jiaqi Bai, Tongliang Li, Qiyao Peng, and 1 others. 2025. xcot: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23550–23558.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. 2024. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7001–7016.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Guanzhi Deng, Bo Li, Ronghao Chen, Xiujin Liu, Zhuo Han, Huacan Wang, Lijie Wen, and Linqi Song. 2026. Dr-lora: Dynamic rank lora for fine-tuning mixture-of-experts models. *Preprint*, arXiv:2601.04823.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

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

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Rongzhi Li and Hitomi Yanaka. 2026. Neuronmoe: Neuron-guided mixture-of-experts for efficient multilingual llm extension. *Preprint*, arXiv:2603.05046.

Zhongyang Li, Ziyue Li, and Tianyi Zhou. 2025. Routing manifold alignment improves generalization of mixture-of-experts llms. *Preprint*, arXiv:2511.07419.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Danni Liu and Jan Niehues. 2025. Middle-layer representation alignment for cross-lingual transfer in fine-tuned llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15979–15996.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, and 1 others. 2025. Olmoe: Open mixture-of-experts language models. In *International Conference on Learning Representations*, volume 2025, pages 62061–62121.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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. *arXiv preprint arXiv:1701.06538*.

Qwen Team. 2024. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters".

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants. https://huggingface.co/datasets/teknium/OpenHermes-2.5.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, volume 2024, pages 45040–45061.

Hongbin Zhang, Kehai Chen, Xuefeng Bai, Yang Xiang, and Min Zhang. 2026. Lingualift: An effective two-stage instruction tuning framework for low-resource language reasoning. *IEEE Transactions on Audio, Speech and Language Processing*.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025. Less, but better: Efficient multilingual expansion for llms via layer-wise mixture-of-experts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17948–17963.

Kening Zheng, Wei-Chieh Huang, Jiahao Huo, Zhonghao Li, Henry Peng Zou, Yibo Yan, Xin Zou, Jungang Li, Junzhuo Li, Hanrong Zhang, Xuming Hu, and Philip S. Yu. 2026. Unveiling language routing isolation in multilingual moe models for interpretable subnetwork adaptation. *Preprint*, arXiv:2604.03592.

Hao Zhou, Zhijun Wang, Shujian Huang, Xin Huang, Xue Han, Junlan Feng, Chao Deng, Weihua Luo, and Jiajun Chen. 2025. Moe-lpr: Multilingual extension of large language models through mixture-of-experts with language priors routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26092–26100.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Data Construction

A.1 Training and Evaluation Datasets

Mathematical reasoning. We sample 60,000 examples from MetaMathQA (Yu et al., 2024), a dataset of rewritten and augmented GSM8K and MATH problems designed to improve mathematical reasoning. Models are evaluated on the GSM8K test set (Cobbe et al., 2021) using exact-match accuracy with an 8-shot prompt.

Instruction following. We sample 60,000 examples from the OLMoE SFT Mix (Muennighoff et al., 2025), a diverse instruction-following corpus. Models are evaluated on IFEval (Zhou et al., 2023) using prompt-level loose accuracy with a 0-shot prompt.

General knowledge. We sample 60,000 examples from OpenHermes-2.5 (Teknium, 2023), a high-quality instruction-tuning corpus covering broad factual and reasoning tasks. Models are evaluated on MMLU (Hendrycks et al., 2021) using 5-shot accuracy.

Task	Train set	Eval set	Metric	Few-shot	Train size
Math Reasoning	MetaMathQA	GSM8K	Accuracy	8-shot	60,000
Instruction Follow.	Tulu 3 SFT Mix	IFEval	Loose acc.	0-shot	60,000
General Knowledge	OpenHermes-2.5	MMLU	Accuracy	5-shot	60,000

Table 4: Training and evaluation dataset pairings for the three downstream tasks.

A.2 Translation Pipeline

Translation model. We translate all English training examples into six target languages (Arabic, Bengali, Chinese, French, Japanese, Spanish) using `google/gemini-2.0-flash-001` accessed via the OpenRouter API, with a zero-shot system prompt instructing the model to produce a faithful translation while preserving the original format.

Dataset-specific handling. Some datasets require targeted adaptations beyond the standard translation pipeline. For MetaMathQA, numerical

expressions, equations, and chain-of-thought answer templates are preserved verbatim during translation to avoid arithmetic errors. For IFEval, since the original verifier (Zhou et al., 2023) assumes English punctuation conventions, we implement per-language adaptations covering two constraint types: sentence boundary detection is updated to each language's native sentence-ending punctuation (e.g., the Arabic full stop and question mark, Bengali Devanagari danda, Chinese and Japanese ideographic period, and standard Latin punctuation for French and Spanish), and comma detection is extended to cover each script's native comma alongside the Latin comma (including the Arabic comma, Japanese ideographic comma, and full-width comma). For MMLU, we bypass automatic translation entirely by directly using the existing multilingual dataset[1], which provides professionally translated test sets for all six target languages.

Translation quality. We evaluate translation quality using COMETkiwi (Rei et al., 2022), a reference-free neural quality estimation model, on all translated splits for each dataset–language pair. Table 5 reports the results. Scores range from 0.80 to 0.88 across all dataset–language pairs, indicating consistently high translation quality. We note that translation errors may introduce spurious *ci* examples into the training signal—cases where the model fails in the target language due to mistranslation rather than a genuine language-driven gap. Such noise would increase the difficulty of the routing alignment task; the fact that RA-MoE consistently outperforms baselines under these conditions therefore speaks to the robustness of our method.

Dataset	ar	bn	zh	fr	ja	es
MetaMathQA	0.83	0.88	0.85	0.87	0.87	0.86
OLMoE-SFT-Mix	0.84	0.87	0.86	0.87	0.87	0.86
OpenHermes-2.5	0.81	0.85	0.84	0.86	0.87	0.86
GSM8K	0.82	0.88	0.85	0.87	0.87	0.86
IFEval	0.80	0.85	0.82	0.84	0.85	0.84

Table 5: COMETkiwi translation quality scores for all translated datasets across six target languages. Higher is better (maximum score: 1.0).

A.3 Correctness Annotation

We judge answer correctness using task-specific procedures applied identically to English and target-language model outputs to ensure consistent *cc*/*ci*/*ic*/*ii* labelling across all three training datasets.

Mathematical reasoning (MetaMathQA). We run greedy decoding on both the English and target-language versions of each training example, and extract the final numerical answer from the model's output using a regex-based extractor that handles multiple answer formats (e.g., #### delimiters, `\boxed{}`, and language-specific answer phrases). An example is marked correct if the extracted answer matches the gold answer after numeric normalization.

Instruction following (OLMoE SFT Mix) and general knowledge (OpenHermes-2.5). For open-ended instruction-following and knowledge data, exact-match verification is not applicable. Instead, we use per-sample perplexity (PPL) over assistant response tokens as a proxy for model competence, with the user prompt tokens excluded. For each parallel pair, we compute the English PPL ($\mathrm{PPL}_{\mathrm{en}}$) and the target-language PPL ($\mathrm{PPL}_{\mathrm{tgt}}$), and define $\Delta = \mathrm{PPL}_{\mathrm{tgt}} - \mathrm{PPL}_{\mathrm{en}}$ as the cross-lingual difficulty gap. Samples with $\mathrm{PPL}_{\mathrm{en}}$ below its corpus-level median are treated as "English-correct"; samples with Δ above its corpus-level median are treated as "target-incorrect", yielding the same four-way *cc*/*ci*/*ic*/*ii* taxonomy. Samples whose PPL values are `NaN` (sequences truncated to zero assistant tokens) or above the 99th percentile of the corpus distribution are excluded before computing the thresholds to avoid instability from degenerate short responses.

B Experimental Implementation

B.1 Model Details

We evaluate RA-MoE on three MoE models spanning different architectures and parameter scales. Table 6 summarises their key architectural properties.

Model	Total params	Active params	Layers	MoE layers	Routed experts	Top-k
OLMoE-1B-7B	7B	1B	16	16	64	8
Qwen1.5-MoE	14.3B	2.7B	24	24	$60+2^{\dagger}$	$4+2^{\dagger}$
DS-V2-Lite	16B	2.4B	27	26	$64+2^{\dagger}$	$6+2^{\dagger}$

Table 6: Architectural summary of the three MoE models. †Qwen1.5-MoE and DS-V2-Lite each include 2 shared experts that are activated for every token in addition to the top-k routed experts. DS-V2-Lite has 1 dense layer followed by 26 MoE layers.

[1] https://huggingface.co/datasets/openai/MMMLU

Model variant selection. A key requirement for RA-MoE is that the chosen model already achieves strong English task performance, so that English routing patterns provide a reliable supervisory signal for target-language inputs. We therefore select model variants based on their English–target gap, quantified by Δ_{avg} (average gap over all six target languages) and Δ_{min} (gap relative to the strongest non-English language). A large gap indicates that the performance deficit is language-driven rather than knowledge-driven, leaving meaningful headroom for routing alignment to operate.

As shown in Table 7, the base versions of OLMoE-1B-7B and DeepSeek-V2-Lite exhibit only marginal English–target gaps across benchmarks. On IFEval in particular, Δ_{avg} drops to 2.5 and 5.8 respectively, suggesting that their deficiencies are not attributable to routing misalignment. We therefore adopt **OLMoE-1B-7B-SFT** and **DeepSeek-V2-Lite-Chat**, whose larger gaps (e.g., $\Delta_{\text{avg}} = 34.6$ on GSM8K for OLMoE-SFT and $\Delta_{\text{avg}} = 28.4$ for DS-V2-Lite-Chat) confirm that English routing patterns are both strong and informative. For Qwen1.5-MoE, the base model already achieves high English performance with substantial gaps across all three tasks (Δ_{avg} up to 62.4 on GSM8K), so we use **Qwen1.5-MoE-A2.7B** directly without instruction tuning.

B.2 Training Hyperparameters

All fine-tuning experiments use LoRA (Hu et al., 2022) applied to the up and down projections of every FFN layer, with all other parameters frozen. All training-based methods (SFT, RISE, RA-MoE) share identical training schedules, optimization settings, and data orders for a fair comparison. Table 8 reports the hyperparameter settings shared across all models and tasks; Table 9 reports the per-model, per-task settings that vary.

B.3 Computational Infrastructure

All experiments are conducted on a single server with 8× RTX 5880 Ada (48 GB each) GPUs. We use DeepSpeed ZeRO-2 for distributed training with bfloat16 mixed precision and Flash Attention 2 for memory-efficient attention computation.

B.4 Evaluation Settings

All evaluations are conducted using vLLM for efficient inference with greedy decoding. We report mean accuracy over 5 independent runs with random seeds $\{42, 123, 456, 789, 2026\}$. Table 10 summarises the per-benchmark evaluation configuration.

B.5 Baseline Reimplementation Details

Routing Steering (Bandarkar et al., 2026) is applied at inference time following the method described in the original paper. Router logits in the middle layers are steered toward the English task-expert activation pattern, applied to the same $\mathcal{L}_{\text{mid}}$ identified in Stage 2 of RA-MoE.

RISE (Zheng et al., 2026) is reproduced following the method described in the original paper. Language-specific expert subnetworks are identified via the language isolation score; only above-median experts are updated during fine-tuning while all other parameters remain frozen. Both baselines use the same training data, learning rate, and schedule as RA-MoE for a fair comparison.

C Computational Cost Analysis

We analyze the computational overhead of RA-MoE along three dimensions: wall-clock training time, GPU memory, and floating-point operations (FLOPs).

C.1 Training Time Analysis

RA-MoE introduces two preparatory stages prior to fine-tuning: Stage 1 annotates the training corpus via vLLM autoregressive decoding, and Stage 2 profiles per-layer routing distributions via teacher-forcing inference. Both stages involve forward passes only, with no gradient computation, and thus run substantially faster than training. Stage 3 fine-tuning adds a scalar KL-divergence term over already-computed router logits, which does not introduce any additional forward or backward operations.

Table 11 reports wall-clock time (hours) for all three models across three downstream tasks, measured on the 8×RTX 5880 Ada server using 4 GPUs per run. As shown, Stage 3 of RA-MoE is on par with standard SFT in every setting, while the cumulative Stage 1+2 overhead amounts to at most 15% of Stage 3 when Stage 2 is re-run independently for each target language. This overhead can be further reduced by exploiting the cross-lingual transferability of intermediate-layer task experts established in Section 4.4.4. When these experts are shared across languages, Stage 2 degenerates to a single English teacher-forcing pass over the *ci* subset, bringing the Stage 1+2 overhead below 10% of Stage 3.

	GSM8K									IFEval									MMLU								
Model	**en**	**ar**	**bn**	**zh**	**fr**	**ja**	**es**	Δ_{avg}	Δ_{min}	**en**	**ar**	**bn**	**zh**	**fr**	**ja**	**es**	Δ_{avg}	Δ_{min}	**en**	**ar**	**bn**	**zh**	**fr**	**ja**	**es**	Δ_{avg}	Δ_{min}
Qwen1.5-MoE-A2.7B	88.5	29.3	3.2	28.8	41.8	9.4	44.4	**62.4**	**44.1**	38.9	18.8	18.9	20.5	28.9	22.9	28.4	**15.8**	**10.0**	54.4	35.8	24.3	46.7	45.2	37.9	45.4	**15.2**	**7.7**
OLMoE-1B-7B-Base	13.0	2.3	1.3	4.2	6.4	3.7	7.5	8.8	5.5	14.8	10.8	8.8	12.1	14.4	13.3	14.2	2.5	0.4	45.5	23.2	26.4	28.2	27.3	25.3	31.3	**18.6**	**14.2**
OLMoE-1B-7B-SFT	47.0	3.8	1.9	17.7	20.7	7.2	23.0	**34.6**	**24.0**	27.9	13.7	11.7	13.8	23.7	17.5	23.1	**10.7**	**4.2**	49.7	28.3	25.5	31.7	37.6	29.6	36.9	18.1	12.1
DS-V2-Lite	36.1	15.2	3.3	32.8	31.7	23.5	33.0	12.9	3.1	20.4	12.8	9.2	13.9	18.0	16.4	17.5	5.8	2.4	39.1	27.7	25.5	37.7	35.8	33.5	37.0	6.2	1.4
DS-V2-Lite-Chat	72.3	26.0	10.2	63.7	58.7	43.7	61.4	**28.4**	**8.6**	32.2	14.4	11.7	15.1	27.4	19.5	26.5	**13.1**	**4.8**	53.9	32.1	28.8	49.9	44.7	40.8	47.4	**13.3**	**4.0**

Table 7: Zero-shot performance (%) of all evaluated model variants across three benchmarks and seven languages. Each condition is averaged over 5 independent runs; standard deviations are omitted for brevity (all $\leq$0.6%). Δ_{avg} is the average gap between English and each of the six target languages; Δ_{min} is the gap between English and the highest-scoring non-English language. Shaded rows mark the variants selected for our experiments.

Hyperparameter	**Value**
LoRA settings	
LoRA rank r	16
LoRA α	32
LoRA dropout	0.05
Target modules	up, down projections
Optimization	
Optimizer	AdamW
Learning rate	2e-5
LR scheduler	Linear
Warmup ratio	0.03
Weight decay	0.0
Gradient clipping	None
Training	
Epochs	1
Training samples	60,000
Precision	bfloat16
RA-MoE specific	
Alignment weight λ	1.0
Task experts per layer K	8
Divergence threshold percentile q	50
General corpus (Stage 2)	FLORES-200
Routing profiling batch size	32

Table 8: Hyperparameter settings shared across all models and tasks.

		MetaMathQA	**OLMoE SFT Mix**	**OpenHermes**
OLMoE	Micro-batch size	16	8	16
	Gradient accum. steps	4	8	4
	Effective batch size	256	256	256
	Max sequence length	512	2048	1024
DS-V2-Lite	Micro-batch size	4	1	2
	Gradient accum. steps	4	8	4
	Effective batch size	64	32	32
	Max sequence length	512	2048	1024
Qwen1.5-MoE	Micro-batch size	2	1	2
	Gradient accum. steps	4	8	4
	Effective batch size	32	32	32
	Max sequence length	512	2048	1024

Table 9: Per-model, per-task batch size and sequence length settings.

Task	**Benchmark**	**Metric**	**Few-shot**	**Test size**
Math Reasoning	GSM8K	Exact match acc.	8-shot	1,319
Instruction Follow.	IFEval	Prompt strict acc.	0-shot	541
General Knowledge	MMLU	Accuracy	5-shot	14,042

Table 10: Evaluation settings for all benchmarks.

		GSM8K		**IFEval**		**MMLU**	
Model	**Method**	**S1+2**	**S3**	**S1+2**	**S3**	**S1+2**	**S3**
OLMoE	SFT	—	3.0	—	6.5	—	3.5
	RA-MoE†	0.5	3.0	1.0	6.5	0.5	3.5
	RA-MoE‡	0.3	3.0	0.5	6.5	0.3	3.5
Qwen1.5	SFT	—	15.5	—	23.0	—	18.5
	RA-MoE†	2.0	15.0	3.5	23.0	3.0	18.5
	RA-MoE‡	1.0	15.0	2.0	23.0	1.5	18.5
DSV2	SFT	—	12.0	—	18.0	—	18.0
	RA-MoE†	1.5	11.5	2.5	18.0	3.0	18.0
	RA-MoE‡	0.8	11.5	1.5	18.0	1.5	18.0

Table 11: Wall-clock time (hours) decomposed by stage. S1+2: Stage 1 + Stage 2; S3: Stage 3. †Stage 2 is re-run independently for each target language. ‡Stage 2 is run once and its intermediate-layer task experts are shared across all target languages.

C.2 Memory Analysis

RA-MoE makes no architectural changes to the underlying model during Stage 3: it applies LoRA at the same rank, to the same modules, as standard SFT. The routing-alignment loss operates solely on the router logits that are already materialized during the forward pass and introduces neither auxiliary parameters nor additional cached activations. Consequently, RA-MoE incurs zero additional GPU memory overhead relative to SFT at the same LoRA configuration.

C.3 FLOPs Analysis

For a forward pass over B tokens, the base MoE computation across L layers, K activated experts per token, hidden dimension d_m, and expert intermediate dimension d_e contributes:

$$\text{FLOPs}_{\text{base}} = 6BLK \cdot d_m \cdot d_e \tag{9}$$

For OLMoE, the factor 6 comes from the three expert projections `gate_proj`, `up_proj`, and `down_proj`. The fused `gate_up_proj` implementation does not change the FLOP count.

LoRA adaptation on the `up_proj` and `down_proj` of each expert (rank r, two modules

per expert) adds:

$$\text{FLOPs}_{\text{LoRA}} = 4BLKr(d_m + d_e) \quad (10)$$

The routing-alignment loss computes a per-layer KL divergence over the router softmax output (dimension E):

$$\text{FLOPs}_{\text{align}} = 2BL \cdot E \quad (11)$$

Since $E \ll d_m d_e$ (e.g., E=64 versus $d_m d_e \approx 2\times10^6$ for OLMoE), $\text{FLOPs}_{\text{align}}$ is negligible. Table 12 instantiates these formulae for OLMoE (B=4096, L=16, K=8, d_m=2048, d_e=1024, r=16, E=64).

Method	Base (GFLOPs)	LoRA (GFLOPs)	Total (GFLOPs)
SFT (no LoRA)	6597.1	—	6597.1
SFT	6597.1	103.1	6700.1
RA-MoE	6597.1	103.1	6700.1

Table 12: FLOPs per forward pass (GFLOPs) for OLMoE (B=4096, L=16, K=8, d_m=2048, d_e=1024, r=16). Base expert computation dominates (98.5%), making the LoRA contribution modest. RA-MoE's routing-alignment loss ($\text{FLOPs}_{\text{align}} < 0.001\%$) is omitted from the table as it is negligible.

As base expert computation dominates total FLOPs (98.5%), RA-MoE and SFT are computationally indistinguishable at Stage 3. The modest LoRA overhead (1.6% over no-LoRA SFT) is identical for both methods, and the alignment loss contributes fewer than $10^{-3}\%$ of total FLOPs.