RA-MoE: Routing-Aligned Fine-Tuning for Multilingual Adaptation of Mixture-of-Experts Models

Anonymous authors
Paper under double-blind review

Abstract

Mixture-of-Experts (MoE) models enable efficient LLM scaling, yet adapting them to non-English downstream tasks remains challenging. Standard multilingual fine-tuning largely ignores their heterogeneous routing structure. Across multiple MoE models and tasks, we find strong cross-lingual routing alignment in middle layers, with routing divergence associated with target-language performance gaps. Motivated by this observation, we propose **RA-MoE** (Routing-Aligned MoE Fine-Tuning), a three-stage framework for multilingual MoE adaptation. RA-MoE categorizes parallel examples into four correctness groups (*cc*/*ci*/*ic*/*ii*) and identifies task-relevant experts in middle layers. It then selectively aligns target-language routing on *ci* examples toward successful English routing patterns, jointly matching the total routing mass assigned to task experts and its relative allocation among them. Experiments across three MoE models, three downstream tasks, and six target languages show that RA-MoE consistently outperforms standard SFT and strong routing-aware baselines. Further analyses confirm the intended routing changes and reveal that middle-layer task routing is largely shared and transferable across languages, providing mechanistic evidence for the cross-language transferability of task-specific routing. Code is available here.

1 Introduction

Mixture-of-Experts (MoE) has emerged as a 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 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 largely inherited from dense LLMs and treat the model as a monolithic learner, without explicitly exploiting the heterogeneous routing structure that MoEs develop during pretraining.

A key clue lies in the routing dynamics of pretrained MoEs. Prior work has revealed a characteristic U-shaped layer-wise pattern in cross-lingual routing divergence: early and late layers tend to be more language-specific, whereas middle layers exhibit substantially stronger cross-lingual alignment (Bandarkar et al., 2026; Chen et al., 2026; Zheng et al., 2026). We find that this structural pattern also persists in task-specific settings.

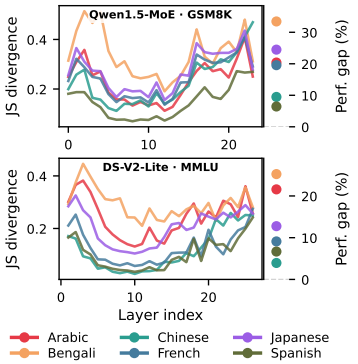


Figure 1: Layer-wise cross-lingual routing divergence. Right-side markers show performance gaps from English.

As shown in Figure 1, the Jensen–Shannon divergence between English and non-English routing distributions is consistently higher in early and late layers and substantially reduced in the middle layers across representative models and downstream tasks. Moreover, languages with larger middle-layer routing divergence tend to exhibit larger task performance gaps relative to English. These observations suggest that middle layers constitute a particularly promising region for cross-lingual routing intervention: they exhibit relatively language-shared routing

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

structure, while residual divergence in this region is associated with degraded target-language task performance.

This motivates a natural question: *can successful routing behavior in English be used as an explicit training signal for adapting MoE models to non-English tasks?* We propose **RA-MoE** (**R**outing-**A**ligned **MoE** Fine-Tuning), a three-stage framework. First, we run inference on parallel task examples $(x_{\text{en}}, x_{\text{tgt}})$ and categorize them as *cc*/*ci*/*ic*/*ii* according to correctness in English and the target language. Second, we identify task-relevant experts in the middle layers by contrasting routing behavior on successful English task examples with that on general English data. Third, during target-language SFT, we selectively align the routing of *ci* examples toward their successful English counterparts, matching both the total routing mass assigned to task experts and its relative allocation among them.

Experiments across three MoE models, three downstream tasks, and six target languages show that RA-MoE consistently outperforms standard SFT and strong routing-aware baselines. Further analyses confirm that these gains are accompanied by the intended routing changes and reveal that task-specific middle-layer routing is largely shared and transferable across languages: successful non-English routing can also provide useful supervision, while task-expert structures identified for one language can be reused for others.

Our contributions are threefold: (1) We validate the characteristic U-shaped cross-lingual routing divergence pattern in task-specific settings and show that middle-layer divergence is associated with target-language performance gaps. (2) We propose **RA-MoE**, which combines correctness-based sample selection, middle-layer task-expert identification, and selective routing alignment that jointly matches task-expert routing mass and allocation. (3) Across multiple MoE models, tasks, and languages, we demonstrate consistent improvements over standard SFT and strong routing-aware baselines, and further show that the underlying task-specific routing structure is largely shared and transferable across languages.

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), sometimes augmented with cross-lingual representation alignment objectives (Liu & Niehues, 2025). These methods improve non-English performance but are primarily 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 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 & Yanaka, 2026). These methods use MoE to augment dense models with multilingual capacity, whereas we focus on improving downstream multilingual performance of models that are already MoE-based.

Multilingual routing analysis in MoEs. Recent studies establish a U-shaped layer-wise pattern in cross-lingual routing divergence, where middle layers exhibit stronger cross-lingual alignment and their routing divergence correlates 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., 2026) shows that existing MoE routers can be suboptimal on downstream tasks 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 uses successful cross-lingual routing patterns as selective supervision for non-English downstream adaptation. RA-MoE bridges this gap by identifying task-relevant experts in the middle layers and selectively aligning target-language routing toward successful English patterns on examples exhibiting cross-lingual correctness asymmetry.

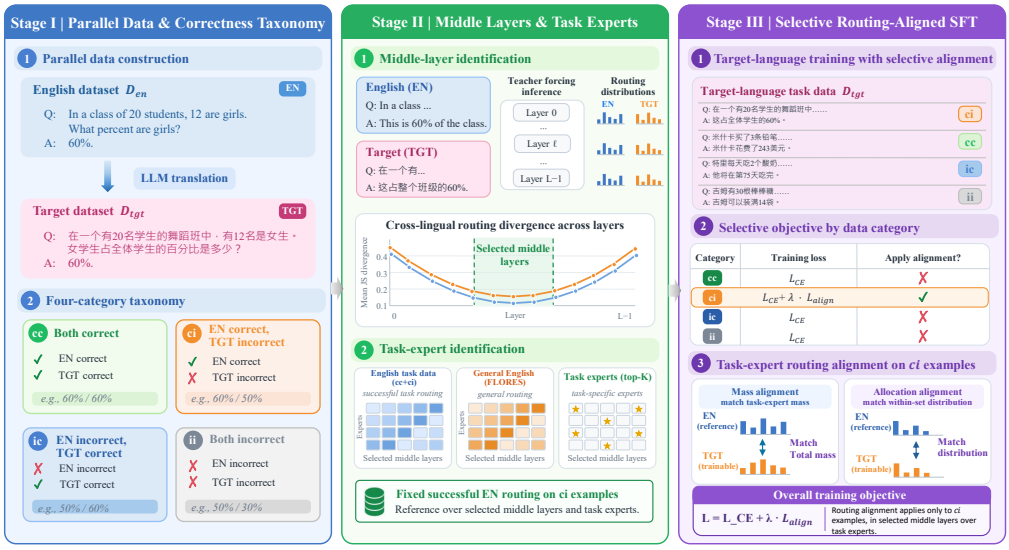


Figure 2: Overview of the RA-MoE framework.

3 METHOD

3.1 OVERVIEW

RA-MoE builds on the task-specific cross-lingual routing pattern established in Figure 1, where routing is substantially more aligned across languages in the middle MoE layers, making these layers a natural target for cross-lingual routing adaptation. As illustrated in Figure 2, RA-MoE proceeds in three stages. **Stage 1** constructs parallel English–target-language task data and partitions examples into four categories according to answer correctness in the two languages. **Stage 2** profiles routing behavior to identify the middle layers with stronger cross-lingual alignment and the task-relevant experts within them. **Stage 3** performs target-language supervised fine-tuning while selectively aligning the routing of *ci* examples toward their successful English counterparts, accounting for both the total routing mass assigned to task experts and its normalized allocation within the task-expert set.

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 construct its target-language counterpart $\mathcal{D}_{\text{tgt}} = \{(x_{\text{tgt}}^{(i)}, y_{\text{tgt}}^{(i)})\}_{i=1}^{N}$, yielding parallel examples $\{(x_{\text{en}}^{(i)}, x_{\text{tgt}}^{(i)})\}_{i=1}^{N}$.

Four-way correctness categorization. We run inference with the MoE model $\mathcal{M}$ on both $x_{\text{en}}^{(i)}$ and $x_{\text{tgt}}^{(i)}$ and evaluate the resulting responses $\hat{y}_{\text{en}}^{(i)}$ and $\hat{y}_{\text{tgt}}^{(i)}$ against their gold answers. Each parallel example is assigned to one of four categories: ***cc*** (both correct), ***ci*** (English correct, target incorrect), ***ic*** (English incorrect, target correct), or ***ii*** (both incorrect). Of particular interest are *ci* examples, which exhibit a clear cross-lingual correctness asymmetry: the model successfully solves the same underlying task in English but fails in the target language. We use these examples as the targets for routing alignment in Stage 3.

3.3 STAGE 2: MIDDLE LAYER AND TASK EXPERT IDENTIFICATION

Routing profiling. We profile routing on the model generations obtained in Stage 1. For each parallel example, we run teacher-forcing inference on $(x_{\text{en}}^{(i)}, \hat{y}_{\text{en}}^{(i)})$ and $(x_{\text{tgt}}^{(i)}, \hat{y}_{\text{tgt}}^{(i)})$ and record the post-softmax router distribution at each MoE layer. For layer ℓ, let $\mathbf{p}_t^{(\text{lang},\ell,i)} \in [0,1]^E$ denote the routing distribution over E experts for token t. We average over generated token positions to obtain

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

where $G^{(i)}_{\text{lang}}$ denotes the generated token positions for example i in language lang.

Middle layer identification. We compute the mean cross-lingual routing divergence at each layer:

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

where D_{JS} denotes the entropy-normalized Jensen–Shannon divergence. Let τ denote the median of $\{\overline{\text{Div}}^{(\ell)}\}_\ell$. We define $\mathcal{L}_{\text{mid}} = \{m, \ldots, n\}$ as the longest contiguous range where $\overline{\text{Div}}^{(\ell)} \leq \tau$ for every $\ell \in \mathcal{L}_{\text{mid}}$. This adaptive criterion selects the low-divergence region illustrated in Figure 1.

Task expert identification. Within each $\ell \in \mathcal{L}_{\text{mid}}$, we identify experts preferentially activated when the model successfully solves the downstream task in English. We contrast correctly answered English task examples $\mathcal{D}^{\text{task}}_{\text{en}}$, corresponding to the *cc* and *ci* subsets, with a general English corpus $\mathcal{D}^{\text{gen}}_{\text{en}}$ (FLORES-200; Costa-Jussà et al. 2022). For each source, the mean routing weight of expert e is

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

and its task-specificity score is

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

We select the top-K experts with $\Delta_e^{(\ell)} > 0$ as the task-expert set $\mathcal{E}^{\text{task}}_\ell$. In our experiments, we use $K = 8$ and analyze sensitivity to K in Section 4.3.

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

3.4 Stage 3: Routing-Aligned SFT

Routing alignment. For each *ci* example i and middle layer $\ell \in \mathcal{L}_{\text{mid}}$, we align the target-language routing with the corresponding successful English routing over the identified task experts. Specifically, we obtain the target-language routing distribution from the current SFT forward pass by averaging router probabilities over response tokens, while keeping the English routing distribution stored in Stage 2 fixed.

Let the total routing mass assigned to the task-expert set be

$$m^{(\text{lang},\ell,i)} = \sum_{e \in \mathcal{E}^{\text{task}}_\ell} q_e^{(\text{lang},\ell,i)}. \tag{5}$$

We construct a $(|\mathcal{E}^{\text{task}}_\ell| + 1)$-dimensional routing distribution that preserves the probability assigned to each task expert while aggregating all remaining experts into a residual component:

$$\mathbf{r}^{(\text{lang},\ell,i)} = \left[\left(q_e^{(\text{lang},\ell,i)}\right)_{e \in \mathcal{E}^{\text{task}}_\ell},\ 1 - m^{(\text{lang},\ell,i)}\right]. \tag{6}$$

The routing alignment loss averages the divergence across *ci* examples and selected middle layers:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{i \in \mathcal{B}_{\text{ci}}} \mathbb{E}_{\ell \in \mathcal{L}_{\text{mid}}} \left[D_{\text{KL}}\left(\mathbf{r}^{(\text{en},\ell,i)} \,\middle\|\, \mathbf{r}^{(\text{tgt},\ell,i)}\right)\right], \tag{7}$$

where $\mathcal{B}_{\text{ci}}$ denotes the *ci* examples in the current minibatch. The explicit minibatch estimator and implementation details are provided in Appendix C.

To characterize this objective, define the normalized task-expert allocation $\tilde{q}_e^{(\text{lang},\ell,i)} = q_e^{(\text{lang},\ell,i)}/m^{(\text{lang},\ell,i)}$ and the binary mass distribution $\mathbf{b}^{(\text{lang},\ell,i)} = [m^{(\text{lang},\ell,i)}, 1 - m^{(\text{lang},\ell,i)}]$. The KL divergence decomposes as

$$D_{\text{KL}}\Big(\mathbf{r}^{(\text{en},\ell,i)} \| \mathbf{r}^{(\text{tgt},\ell,i)}\Big) = D_{\text{KL}}\Big(\mathbf{b}^{(\text{en},\ell,i)} \| \mathbf{b}^{(\text{tgt},\ell,i)}\Big) + m^{(\text{en},\ell,i)} D_{\text{KL}}\Big(\tilde{\mathbf{q}}^{(\text{en},\ell,i)} \| \tilde{\mathbf{q}}^{(\text{tgt},\ell,i)}\Big). \quad (8)$$

The first term aligns the total routing mass assigned to task experts, while the second aligns the normalized allocation within the task-expert set, weighted by the English task-expert mass. A detailed derivation is provided in Appendix B.3.

Combined objective. The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{align}}, \quad (9)$$

where λ controls the strength of routing alignment. Only *ci* examples contribute to $\mathcal{L}_{\text{align}}$, while all training examples contribute to $\mathcal{L}_{\text{CE}}$.

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** (Qwen Team 2024, Qwen1.5), and **DeepSeek-V2-Lite-Chat** (DeepSeek-AI et al. 2024, DSV2).

Tasks and languages. We evaluate on three downstream tasks: **Mathematical Reasoning** (GSM8K; Cobbe et al. 2021), **Logical Reasoning** (LogicBench; Parmar et al. 2024), and **Knowledge Understanding** (MMLU; Hendrycks et al. 2021). For each task, we construct parallel English–target-language data in 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 four baselines: **Zero-shot**, which evaluates the pretrained MoE directly on target-language inputs without task-specific fine-tuning; **Routing Steering** (Bandarkar et al., 2026), an inference-time intervention that promotes English-aligned expert activation in middle layers without updating model parameters; **SFT**, which performs standard supervised fine-tuning on target-language task data; and **RISE** (Zheng et al., 2026), which selectively fine-tunes language-specific expert subnetworks identified through routing analysis, without an explicit cross-lingual alignment signal. Implementation details are provided in Appendix C.

4.2 Main Results

Table 1 presents the main results across three MoE models, three tasks, and six target languages. We report mean accuracy and standard deviation over five evaluation runs using different random seeds.

Several observations emerge. First, RA-MoE consistently improves over standard SFT, outperforming it in 53 of the 54 model–task–language settings and tying in the remaining one, with an average gain of 2.4 percentage points. The improvement holds across all three tasks: RA-MoE gains 2.1 points on GSM8K, 3.1 on LogicBench, and 1.9 on MMLU on average. In particular, it improves over SFT in all 18 LogicBench and all 18 MMLU settings, while achieving 17 wins and one tie on GSM8K.

Second, RA-MoE substantially outperforms Routing Steering across all 54 settings, with an average gain of 7.0 points. While Routing Steering directly modifies expert routing at inference time, its improvements over the zero-shot model are limited in many cases. In contrast, incorporating routing supervision during fine-tuning yields substantially stronger adaptation across reasoning, logical, and knowledge-intensive tasks.

Table 1: Main results (%) across three tasks, three models, and six target languages. 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.

		GSM8K						LogicBench						MMLU						
Model	**Method**	**ar**	**bn**	**zh**	**fr**	**ja**	**es**	**ar**	**bn**	**zh**	**fr**	**ja**	**es**	**ar**	**bn**	**zh**	**fr**	**ja**	**es**	**Avg.**
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	33.7 ±0.5	15.2 ±0.4	43.5 ±1.2	41.2 ±0.6	41.3 ±0.7	46.2 ±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	26.9 ±0.3
	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	34.3 ±0.4	15.5 ±0.2	43.2 ±0.4	41.6 ±0.3	41.3 ±0.5	46.5 ±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	27.1 ±0.2
	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	34.9 ±0.6	15.7 ±0.4	44.5 ±0.7	42.8 ±0.7	43.5 ±1.1	48.8 ±0.9	28.4 ±0.1	25.6 ±0.1	32.1 ±0.1	38.0 ±0.1	29.7 ±0.1	37.8 ±0.1	29.0 ±0.4
	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	35.7 ±0.8	16.3 ±0.4	45.8 ±0.7	43.6 ±1.0	45.3 ±0.8	50.5 ±0.7	29.1 ±0.1	25.8 ±0.1	32.9 ±0.1	38.2 ±0.1	29.9 ±0.1	38.1 ±0.1	29.6 ±0.4
	RA-MoE	**10.8 ±0.2**	**2.2 ±0.2**	**22.7 ±0.3**	**26.9 ±0.3**	**15.3 ±0.5**	**29.8 ±0.5**	**37.5 ±0.4**	**17.2 ±0.6**	**47.8 ±0.8**	**46.3 ±0.5**	**47.2 ±0.6**	**51.7 ±0.7**	**30.6 ±0.1**	**27.7 ±0.1**	**33.6 ±0.1**	**39.3 ±0.1**	**31.5 ±0.1**	**39.2 ±0.1**	**31.0 ±0.3**
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	59.6 ±0.8	64.4 ±0.3	63.1 ±0.7	53.9 ±1.5	60.6 ±0.7	66.8 ±0.4	35.8 ±0.1	24.3 ±0.1	46.7 ±0.1	45.2 ±0.1	37.9 ±0.1	45.4 ±0.1	42.3 ±0.4
	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	60.2 ±0.6	63.5 ±0.5	63.4 ±0.3	53.5 ±0.8	61.2 ±0.4	66.2 ±0.7	36.4 ±0.1	24.9 ±0.1	46.0 ±0.1	45.8 ±0.1	38.3 ±0.1	45.1 ±0.1	42.4 ±0.4
	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	61.1 ±0.9	65.5 ±0.5	65.2 ±0.9	55.8 ±1.1	62.6 ±1.3	68.5 ±0.6	40.0 ±0.1	28.3 ±0.1	52.8 ±0.1	50.6 ±0.1	42.7 ±0.1	51.2 ±0.1	52.1 ±0.5
	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	62.0 ±0.7	66.1 ±0.4	66.2 ±0.6	56.8 ±1.2	63.7 ±0.8	69.2 ±0.8	40.3 ±0.1	28.6 ±0.1	53.2 ±0.1	51.0 ±0.1	**43.1 ±0.1**	52.3 ±0.1	52.8 ±0.4
	RA-MoE	**52.0 ±0.5**	**32.9 ±0.5**	**59.8 ±0.3**	**60.2 ±0.7**	**46.6 ±0.7**	**61.3 ±0.5**	**63.4 ±0.6**	**67.2 ±1.1**	**67.5 ±0.5**	**58.4 ±0.8**	**65.3 ±0.3**	**70.4 ±0.8**	**42.1 ±0.1**	**30.3 ±0.1**	**55.2 ±0.1**	**52.5 ±0.1**	**43.1 ±0.1**	**53.9 ±0.1**	**54.6 ±0.4**
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	54.6 ±0.7	47.7 ±1.1	57.6 ±1.0	54.8 ±0.6	53.7 ±0.8	53.3 ±0.9	32.1 ±0.1	28.8 ±0.1	49.9 ±0.1	44.7 ±0.1	40.8 ±0.1	47.4 ±0.1	46.1 ±0.5
	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	55.0 ±0.6	48.2 ±0.5	57.1 ±0.5	56.1 ±0.7	53.9 ±0.3	54.3 ±0.2	32.5 ±0.1	28.9 ±0.1	49.3 ±0.1	44.4 ±0.1	41.1 ±0.1	47.9 ±0.1	46.4 ±0.3
	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	55.5 ±0.9	49.1 ±0.6	58.6 ±0.7	58.6 ±1.0	55.9 ±0.9	58.8 ±1.0	32.3 ±0.1	29.2 ±0.1	50.0 ±0.1	45.0 ±0.1	42.2 ±0.1	48.2 ±0.1	48.7 ±0.4
	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	57.0 ±0.8	50.5 ±0.7	60.9 ±1.1	60.2 ±0.7	57.2 ±0.6	61.5 ±0.9	32.9 ±0.1	29.3 ±0.1	50.1 ±0.1	45.2 ±0.1	42.5 ±0.1	48.3 ±0.1	49.5 ±0.4
	RA-MoE	**39.4 ±0.6**	**20.8 ±0.3**	**65.6 ±0.4**	**61.4 ±0.6**	**53.1 ±0.5**	**64.6 ±0.2**	**58.9 ±1.3**	**52.8 ±0.8**	**63.3 ±0.7**	**62.6 ±0.8**	**59.3 ±0.5**	**63.5 ±0.9**	**35.0 ±0.1**	**30.5 ±0.1**	**52.2 ±0.1**	**48.0 ±0.1**	**43.7 ±0.1**	**49.9 ±0.1**	**51.4 ±0.5**

Third, RA-MoE consistently improves over RISE, outperforming it in 53 of 54 settings and tying once, with an average gain of 1.7 points. The advantage is observed across all three tasks: RA-MoE outperforms RISE in every GSM8K and LogicBench setting, and in 17 of 18 MMLU settings. These results indicate that explicitly aligning target-language routing toward successful English task-routing patterns provides gains beyond selectively adapting language-specific expert subnetworks.

4.3 Ablation and Sensitivity Analysis

Ablation. To assess the contribution of each component of RA-MoE, we conduct ablations on Qwen1.5 with GSM8K in Bengali, Arabic, and Spanish. Table 2 compares the full method with six variants: **(1) w/o $\mathcal{L}_{\text{align}}$**, removing routing alignment; **(2) w/o mass alignment** and **(3) w/o allocation alignment**, removing the mass and within-set allocation components, respectively; **(4) w/o task experts**, aligning all experts in the selected middle layers; **(5) w/o *ci* filtering**, aligning all training examples; and **(6) w/o middle layers**, aligning all MoE layers.

As shown in Table 2, all ablations degrade performance relative to the full RA-MoE. Removing routing alignment causes the largest overall drop. Among the two alignment components, retaining only mass alignment consistently outperforms retaining only allocation alignment, while combining both achieves the best results, indicating their complementary roles. The remaining ablations further confirm the importance of selectively aligning task experts, *ci* examples, and middle MoE layers.

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 (GSM8K). RA-MoE remains robust across a broad range of both hyperparameters. Too small a λ weakens the routing alignment signal, whereas an overly large λ can interfere with the cross-entropy objective. Similarly, too small a K may omit relevant experts, while too large a K can include less task-specific experts and dilute the alignment target.

Table 2: Ablation study on Qwen1.5 (GSM8K).

Method	**bn**	**ar**	**es**
w/o $\mathcal{L}_{\text{align}}$	29.5 ±0.6	48.8 ±0.4	58.5 ±0.6
w/o mass alignment	31.8 ±0.6	50.7 ±0.3	60.4 ±0.5
w/o allocation alignment	32.2 ±0.5	51.3 ±0.4	60.6 ±0.6
w/o task experts	31.5 ±0.5	50.5 ±0.3	59.4 ±0.7
w/o *ci* filtering	31.4 ±0.4	50.2 ±0.4	59.3 ±0.5
w/o middle layers	30.2 ±0.6	49.4 ±0.6	58.5 ±0.3
RA-MoE	**32.9 ±0.5**	**52.0 ±0.5**	**61.3 ±0.5**

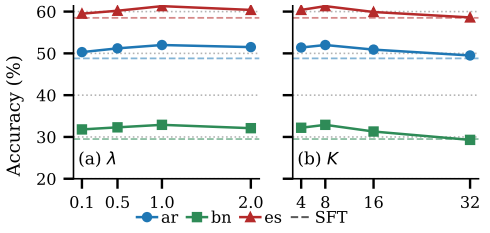


Figure 3: Sensitivity analysis on Qwen1.5 (GSM8K). Dashed lines denote SFT baselines.

5 Further Analysis

We analyze when RA-MoE is most beneficial, whether its gains are accompanied by the intended routing changes, and how broadly the underlying task-routing structure transfers across languages.

5.1 Higher *ci* Proportions Correlate with Larger Gains

We examine whether the proportion of *ci* examples is associated with the benefit of RA-MoE over standard SFT. As shown in Figure 4, the relative gain over SFT is positively correlated with the *ci* proportion across languages and models ($r = 0.70$, $p = 0.001$).

This trend is consistent with the selective design of RA-MoE: *ci* examples capture cases where the model succeeds in English but fails on the same task in the target language, making the successful English routing pattern an informative alignment reference. By contrast, *cc* examples offer less corrective signal, while *ii* examples lack a successful English reference. Thus, task–language pairs with more *ci* examples provide more actionable routing supervision. OLMoE on Bengali is a notable outlier, showing little gain despite a high *ci* proportion, suggesting that this signal is most informative when the base model already has sufficient target-language capability.

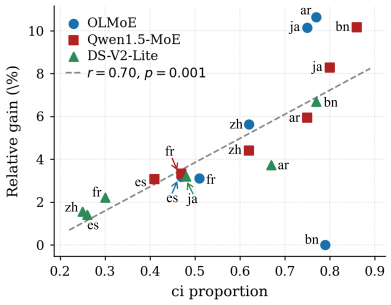
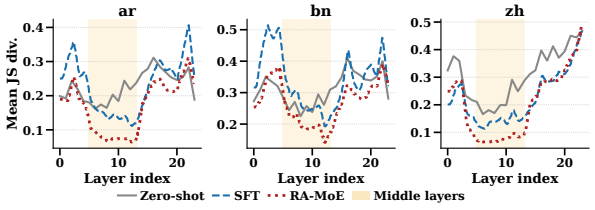


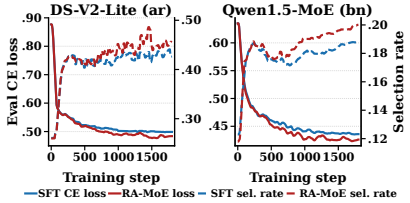
Figure 4: Relative gain of RA-MoE over SFT vs. *ci* proportion on GSM8K.

5.2 RA-MoE Induces the Intended Routing Changes

Having identified when RA-MoE tends to yield larger gains, we next examine whether these gains are accompanied by the intended routing changes. Specifically, we analyze cross-lingual routing divergence before and after fine-tuning and track task-expert activation throughout training.



(a)



(b)

Figure 5: Routing-level analysis of RA-MoE. (a) Layer-wise cross-lingual routing divergence on Qwen1.5 (GSM8K); shading marks selected middle layers. (b) Eval CE loss and task-expert selection rate (fraction of routed top-k slots assigned to task experts) under SFT and RA-MoE.

RA-MoE substantially reduces middle-layer routing divergence. As shown in Figure 5(a), before fine-tuning, all three target languages exhibit pronounced divergence from English, including within the selected middle layers. Standard SFT reduces this divergence only modestly, whereas RA-MoE consistently brings Arabic, Bengali, and Chinese routing closer to the English reference. This shows that the performance improvements of RA-MoE are accompanied by the intended change in cross-lingual routing behavior.

RA-MoE increases task-expert activation while lowering eval CE loss. Figure 5(b) tracks these two signals throughout training on two representative model–language pairs. Under both methods, task-expert activation increases as training progresses, but RA-MoE reaches a consistently higher selection rate. Meanwhile, its eval CE loss becomes lower than that of SFT after the early training stage. Together, these trends suggest that routing alignment encourages stronger engagement of the task-relevant experts identified in Stage 2 while improving target-language task adaptation.

5.3 Task-Specific Routing Transfers Across Languages

Beyond examining how RA-MoE changes routing during adaptation, we next ask whether task-specific routing structure transfers across languages. We study this question from two complementary perspectives: whether successful routing supervision can come from non-English reference languages, and whether the middle-layer task-expert structure identified for one target language can be reused for others.

Table 3: Effect of reference language.

Model	Ref.	Ref. Acc.	Target Acc.	Gain
DSV2	SFT	–	50.0 ±0.4	–
	en	72.3	53.1 ±0.5	+3.1
	zh	64.2	51.7 ±0.4	+1.7
	es	63.4	51.2 ±0.4	+1.2
Qwen1.5	SFT	–	42.3 ±0.4	–
	en	88.5	46.6 ±0.7	+4.3
	fr	57.2	43.8 ±0.5	+1.5
	es	58.5	44.5 ±0.4	+2.2

Table 4: Cross-language transfer of task experts.

		GSM8K		LogicBench		MMLU		Avg. Gain over SFT
Model	Method	ar	bn	ar	bn	ar	bn	
OLMoE	RA-MoE	10.8 ±0.2	2.2 ±0.2	37.5 ±0.4	17.2 ±0.6	30.6 ±0.1	27.7 ±0.1	+1.6
	RA-MoE†	10.2 ±0.2	2.1 ±0.1	36.0 ±0.5	16.6 ±0.4	29.2 ±0.2	26.5 ±0.1	+0.7
Qwen1.5	RA-MoE	52.0 ±0.5	32.9 ±0.5	63.4 ±0.6	67.2 ±1.1	42.1 ±0.1	30.3 ±0.1	+2.5
	RA-MoE†	51.1 ±0.5	31.9 ±0.4	62.5 ±0.6	66.0 ±0.9	40.8 ±0.1	29.0 ±0.1	+1.4
DSV2	RA-MoE	39.4 ±0.6	20.8 ±0.3	58.9 ±1.3	52.8 ±0.8	35.0 ±0.1	30.5 ±0.1	+2.4
	RA-MoE†	38.8 ±0.5	20.2 ±0.2	57.5 ±0.8	51.2 ±0.7	33.3 ±0.1	29.5 ±0.1	+1.2

Successful routing supervision transfers beyond English. We first ask whether English is uniquely required as the routing reference. We consider two GSM8K settings, both targeting Japanese: DSV2 with English, Chinese, or Spanish as the reference, and Qwen1.5 with English, French, or Spanish as the reference. Each non-English reference language is first adapted with standard SFT to ensure sufficient task competence before its routing patterns are used for alignment. In both settings, the non-English reference languages substantially outperform Japanese on the task.

As shown in Table 3, all non-English references improve target-language performance over standard SFT, showing that effective routing supervision is not specific to English. A language with substantially stronger task competence can also serve as a useful reference for a weaker target language. However, non-English references yield smaller gains than English, which has the largest performance advantage over the target language. This suggests that routing-transfer effectiveness is associated with the reference–target task-competence gap.

Task experts can be reused across target languages. We next examine whether the task-routing structure identified for one target language can be transferred to others. For each task, we directly reuse the middle-layer range and task-expert sets identified under the Chinese target-language configuration when fine-tuning Arabic or Bengali. We denote this setting as RA-MoE†.

As shown in Table 4, RA-MoE† preserves positive average gains over SFT across all three models, although its improvements are smaller than those of standard RA-MoE. Thus, middle-layer task-expert structure can be reused across target languages without re-running the identification stage.

5.4 Mechanistic Evidence for Language-Shared Task Routing

The transfer results above show that routing supervision and task-expert structure can both transfer across languages. We next ask what routing structure underlies this transferability. To this end, we compare task experts identified for GSM8K across English, Chinese, and Spanish on DSV2. For routing-pattern comparisons, we use a shared set of parallel examples correctly solved in all three reference languages but incorrectly solved in Japanese, controlling for sample composition.

Task-expert identity is substantially more stable across languages than across tasks. For two task-expert sets $\mathcal{E}_A^{(\ell)}$ and $\mathcal{E}_B^{(\ell)}$ at layer ℓ, we measure their overlap using Jaccard similarity: $J^{(\ell)}(A,B) = \frac{|\mathcal{E}_A^{(\ell)} \cap \mathcal{E}_B^{(\ell)}|}{|\mathcal{E}_A^{(\ell)} \cup \mathcal{E}_B^{(\ell)}|}$. As shown in Figure 6(a), the mean cross-language Jaccard similarity on GSM8K is 0.458 across English, Chinese, and Spanish, compared with only 0.054 between GSM8K and MMLU when the language is fixed to English. The same-task cross-language overlap is therefore about 8.4× higher than the cross-task overlap. It also substantially exceeds the random-expert baseline (0.071 for $E = 64$ and $K = 8$); even the smallest observed cross-language Jaccard (0.410) is matched by the random baseline with an empirical probability of only 3.57×10^{-4}. These results suggest that middle-layer expert identity is considerably more task-dependent than language-dependent.

Task-expert usage is also similar across languages. Figure 6(b) shows that English, Chinese, and Spanish assign very similar mean routing mass to the shared GSM8K task experts (0.184, 0.183, and 0.175, respectively), with mean pairwise differences below 0.011. Figure 6(c) shows that their normalized routing distributions within this shared set are highly similar, with pairwise JS divergence of only 0.0023–0.0033. Thus, successful execution of the same task across languages is characterized not only by overlapping expert identity but also by similar routing patterns over those experts.

Together, these results provide mechanistic evidence for a language-shared view of middle-layer task routing: once the task is fixed, successful execution across languages tends to recruit a common set

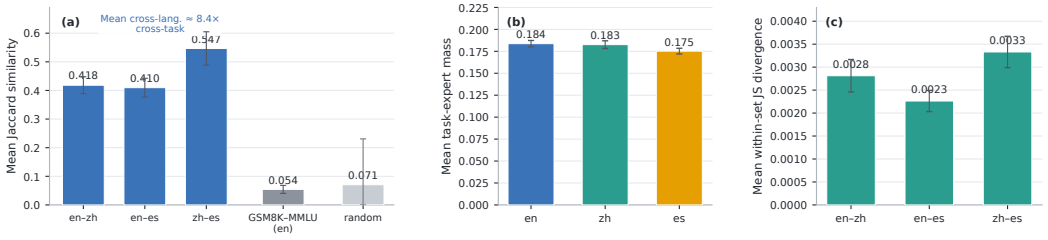


Figure 6: Language-shared middle-layer task routing on DSV2 (GSM8K). (a) Task-expert overlap with cross-task and random controls. (b) Mean task-expert mass. (c) Within-set JS divergence.

of task-relevant experts and use them in similar ways. Cross-lingual performance gaps may therefore arise partly from how effectively each language accesses this shared task-expert structure, rather than from requiring a fundamentally different set of experts.

5.5 Extension to Proxy-Based Sample Selection

Our main experiments focus on tasks for which training examples can be categorized using objective answer correctness, allowing *cc*/*ci*/*ic*/*ii* to be defined without auxiliary quality proxies. We further examine whether RA-MoE can be extended to settings where exact correctness labels are unavailable for training-time sample selection.

Table 5: Proxy-based extension on IFEval.

Model	SFT	RA-MoE	Gain
OLMoE	21.1 ±0.4	**22.8 ±0.4**	+1.7
Qwen1.5	24.8 ±0.4	**27.0 ±0.4**	+2.2
DSV2	24.2 ±0.3	**26.5 ±0.4**	+2.3
Avg.	23.4	**25.4**	+2.1

We consider IFEval, whose downstream performance can still be evaluated objectively, while its open-ended instruction-following training data do not provide per-response correctness labels. We therefore replace exact *ci* labels with a PPL-derived *proxy-ci* criterion. For each parallel pair, we compute perplexity over assistant-response tokens for the English and target-language responses, $\mathrm{PPL}_{\mathrm{en}}$ and $\mathrm{PPL}_{\mathrm{tgt}}$, and define $\Delta = \mathrm{PPL}_{\mathrm{tgt}} - \mathrm{PPL}_{\mathrm{en}}$. An example is selected as *proxy-ci* when $\mathrm{PPL}_{\mathrm{en}}$ is below its corpus-level median and Δ is above its corpus-level median. We use this criterion only as a noisy sample-selection signal rather than a ground-truth correctness label.

RA-MoE with proxy-based selection improves over standard SFT in all 18 model–language settings, with an average gain of 2.1 points as shown in Table 5. This suggests that exact correctness labels are not strictly required when an informative relative-quality signal is available: *ci* selection serves to identify a reliable reference-stronger/target-weaker asymmetry, for which response likelihood can provide an approximate signal. A manual inspection of 600 *proxy-ci* examples across six target languages finds that 85.2% match the intended pattern, while 7.1% are false positives caused by translation or formatting artifacts and 7.7% are ambiguous. Thus, although imperfect, PPL-derived selection captures the intended cross-lingual competence asymmetry in most cases and remains a useful, albeit noisy, alignment signal.

6 Conclusion

We presented RA-MoE, a routing-aligned fine-tuning framework for multilingual adaptation of MoE models. Motivated by the task-specific U-shaped pattern of cross-lingual routing divergence, RA-MoE selectively aligns target-language routing on *ci* examples toward successful English task-expert patterns in the middle layers, jointly matching task-expert routing mass and within-set allocation. Experiments across three MoE models, three downstream tasks, and six target languages show consistent improvements over standard SFT and strong routing-aware baselines. Further analyses confirm the intended routing changes, show that routing supervision and task-expert structure can transfer across languages, and demonstrate that informative proxy signals can extend RA-MoE beyond settings with exact correctness labels. Overall, our findings suggest that successful task execution across languages relies on a largely shared middle-layer routing structure. Future work may explore richer quality signals for open-ended tasks and adaptive selection of reference languages.

AI USE STATEMENT

In this work, generative AI tools were used solely to assist with English writing. All research ideas, experimental design, implementation, and results are the authors' own work. All AI-assisted text was reviewed and edited by the authors, who take responsibility for the final content of the paper.

REFERENCES

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. Multilingual routing in mixture-of-experts. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZoZR0x7tTD.

Linzheng Chai, Jian Yang, Tao Sun, Hongcheng Guo, Jiaheng Liu, Bing Wang, Xinnian Liang, Jiaqi Bai, Tongliang Li, Qiyao Peng, and Zhoujun Li. XCOT: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23550–23558, 2025. doi: 10.1609/aaai.v39i22.34524.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7001–7016, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.411. URL https://aclanthology.org/2024.findings-emnlp.411/.

Yuxin Chen, Zhengzhou Cai, Xiangtian Ji, Weixiang Zhao, An Zhang, Xiang Wang, and Tat-Seng Chua. Understanding multilingualism in mixture-of-experts LLMs: Routing mechanism, expert specialization, and layerwise steering. *arXiv preprint arXiv:2601.14050*, 2026. URL https://arxiv.org/abs/2601.14050.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, et al. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024. URL https://arxiv.org/abs/2405.04434.

Guanzhi Deng, Bo Li, Ronghao Chen, Xiujin Liu, Zhuo Han, Huacan Wang, Lijie Wen, and Linqi Song. DR-LoRA: Dynamic rank LoRA for fine-tuning mixture-of-experts models. *arXiv preprint arXiv:2601.04823*, 2026. URL https://arxiv.org/abs/2601.04823.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. URL https://arxiv.org/abs/2401.04088.

Rongzhi Li and Hitomi Yanaka. NeuronMoE: Efficient cross-lingual extension via neuron-guided mixture-of-experts. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2573–2586, Rabat, Morocco, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-long.117. URL https://aclanthology.org/2026.eacl-long.117/.

Zhongyang Li, Ziyue Li, and Tianyi Zhou. Routing manifold alignment improves generalization of mixture-of-experts LLMs. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3lskwxB653.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. URL https://arxiv.org/abs/2412.19437.

Danni Liu and Jan Niehues. 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)*, pp. 15979–15996, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.778. URL https://aclanthology.org/2025.acl-long.778/.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. OLMoE: Open mixture-of-experts language models. In *International Conference on Learning Representations*, 2025.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13679–13707, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.739. URL https://aclanthology.org/2024.acl-long.739/.

Qwen Team. Qwen1.5-MoE: Matching 7B model performance with 1/3 activated parameters. Qwen Blog, March 2024. URL https://qwenlm.github.io/blog/qwen-moe/.

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. COMETKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 634–645, Abu Dhabi, United Arab Emirates (Hybrid), 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.wmt-1.60. URL https://aclanthology.org/2022.wmt-1.60/.

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

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

Hongbin Zhang, Kehai Chen, Xuefeng Bai, Yang Xiang, and Min Zhang. LinguaLIFT: An effective two-stage instruction tuning framework for low-resource language reasoning. *IEEE Transactions on Audio, Speech and Language Processing*, 34:1578–1593, 2026. doi: 10.1109/TASLPRO.2026.3671636.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 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)*, pp. 17948–17963, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.878. URL https://aclanthology.org/2025.acl-long.878/.

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. Unveiling language routing isolation in multilingual MoE models for interpretable subnetwork adaptation. *arXiv preprint arXiv:2604.03592*, 2026. URL https://arxiv.org/abs/2604.03592.

Hao Zhou, Zhijun Wang, Shujian Huang, Xin Huang, Xue Han, Junlan Feng, Chao Deng, Weihua Luo, and Jiajun Chen. 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, pp. 26092–26100, 2025. doi: 10.1609/aaai.v39i24.34805.

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

A Data Construction

A.1 Core Training and Evaluation Datasets

Our core experiments cover three downstream tasks with objectively verifiable answers: mathematical reasoning, logical reasoning, and knowledge understanding. For each task, we construct parallel English–target-language training examples in Arabic (ar), Bengali (bn), Chinese (zh), French (fr), Japanese (ja), and Spanish (es). Table 6 summarizes the training/evaluation pairings used in the main experiments.

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

Logical reasoning. We use LogicBench (Parmar et al., 2024) for logical reasoning, using LogicBench (Aug) for training and LogicBench (Eval) for evaluation. The task admits objective answer labels, allowing exact correctness-based categorization in Stage 1.

Knowledge understanding. We evaluate on MMLU (Hendrycks et al., 2021) using 0-shot accuracy. For adaptation, we sample 60,000 examples from the MMLU auxiliary/train split as the English training data, while the MMLU test set is reserved for evaluation in the six target languages.

Table 6: Training and evaluation dataset pairings for the three core downstream tasks.

Task	Train set	Eval set	Metric	Few-shot	Train size
Math Reasoning	MetaMathQA	GSM8K	Exact-match acc.	8-shot	60,000
Logical Reasoning	LogicBench Aug-BQA	LogicBench Eval-BQA	Accuracy	0-shot	12,908
Knowledge Understanding	MMLU auxiliary train	MMLU test	Accuracy	0-shot	60,000

A.2 Translation Pipeline

Translation model. We translate English training and evaluation data into the six target languages using `google/gemini-2.5-flash-lite` accessed through the OpenRouter API, where applicable.

Dataset-specific handling. For MetaMathQA, numerical expressions, equations, and chain-of-thought answer templates are preserved verbatim during translation to avoid introducing arithmetic errors. For LogicBench, option labels, symbolic entities, and task-specific constraint templates are kept unchanged to preserve the underlying logical structure and answer space. For GSM8K evaluation, we translate the original English test split into the six target languages using the same translation pipeline. For MMLU, we sample the English adaptation data from the auxiliary/train split and translate them into the six target languages through OpenRouter, while evaluation uses the existing multilingual MMLU benchmark rather than automatically translating the English test set.

Translation quality. We use COMETkiwi (Rei et al., 2022), a reference-free neural quality-estimation model, to assess translated splits. Table 7 reports translation-quality scores for all datasets translated through OpenRouter in our experiments.

Table 7: COMETkiwi translation-quality scores for datasets translated through OpenRouter across six target languages. The **Role** column indicates how each dataset is used in our experiments. Higher is better.

Dataset	Role	ar	bn	zh	fr	ja	es
MetaMathQA	GSM8K train	0.88	0.85	0.88	0.87	0.87	0.86
GSM8K (test)	GSM8K eval	0.83	0.88	0.85	0.87	0.87	0.86
LogicBench (Aug & Eval)	LogicBench train/eval	0.87	0.84	0.87	0.86	0.85	0.85
MMLU (auxiliary/train)	MMLU train	0.88	0.87	0.85	0.87	0.86	0.88
OLMoE-SFT-Mix	IFEval train	0.84	0.87	0.86	0.87	0.87	0.86
IFEval	IFEval eval	0.89	0.85	0.88	0.87	0.85	0.86

A.3 Exact Correctness Annotation for the Core Tasks

The three core tasks use objective correctness criteria to define the *cc*/*ci*/*ic*/*ii* categories. We apply the same task-specific evaluation procedure to the English and target-language inputs for each parallel example.

Mathematical reasoning (MetaMathQA/GSM8K). We run greedy decoding on the English and target-language versions of each training example and extract the final numerical answer using a regex-based extractor supporting common answer formats (e.g., #### delimiters, `\boxed{}`, and language-specific answer phrases). A response is marked correct if its extracted answer matches the gold answer after numeric normalization.

Logical reasoning (LogicBench). For both Aug-BQA and Eval-BQA, the valid prediction space is $\{\texttt{yes}, \texttt{no}\}$. We first search the generated response, case-insensitively, for occurrences matching `Final answer: yes` or `Final answer: no`, allowing arbitrary whitespace around the colon, and use the last occurrence if multiple matches are found. Otherwise, we fall back to the last standalone occurrence of `yes` or `no` in the response. If neither label can be extracted, the prediction is treated as empty and counted as incorrect. English gold labels are stripped and lowercased. For translated Eval-BQA data, language-specific affirmative and negative gold labels are deterministically mapped to the canonical `yes`/`no` labels before comparison using a fixed mapping for Arabic, Bengali, Chinese, French, Japanese, and Spanish. Model responses are evaluated in this canonical label space.

Knowledge understanding (MMLU). Training-time MMLU correctness is computed as a constrained multiple-choice decision over $\{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$ rather than by parsing a free-form generated response. After a language-specific answer prefix, we perform a single-token, zero-temperature decoding step and obtain the next-token log probability for each of the four option-letter tokens. The option with the highest log probability is selected and compared exactly with the gold answer. Integer gold labels, when present, are mapped positionally as $0 \mapsto \texttt{A}$, $1 \mapsto \texttt{B}$, $2 \mapsto \texttt{C}$, and $3 \mapsto \texttt{D}$. No normalization of spelled-out, language-specific, parenthesized, or otherwise free-form answer surfaces is required because prediction is restricted directly to the four option tokens.

A.4 Proxy-Based Sample Selection for IFEval

Section 5.5 examines whether RA-MoE can be extended to settings where exact correctness labels are unavailable for training-time sample selection. We use the OLMoE SFT Mix as open-ended instruction-following training data and evaluate on IFEval (Zhou et al., 2023) using prompt-level loose accuracy. Unlike the three core tasks, this extension uses a response-likelihood proxy to identify examples for routing alignment.

PPL-derived *proxy-ci*. For each parallel training pair, we compute per-sample perplexity over assistant-response tokens only, excluding user-prompt tokens. Let $\mathrm{PPL}_{\mathrm{en}}$ and $\mathrm{PPL}_{\mathrm{tgt}}$ denote the English and target-language perplexities, and define

$$\Delta_{\mathrm{PPL}} = \mathrm{PPL}_{\mathrm{tgt}} - \mathrm{PPL}_{\mathrm{en}}. \tag{10}$$

An example is selected as *proxy-ci* when $\mathrm{PPL}_{\mathrm{en}}$ is below its corpus-level median and Δ_{PPL} is above its corpus-level median. Samples with undefined PPL (e.g., zero valid assistant tokens) or PPL above the 99th percentile are excluded before computing the thresholds. This criterion is intended to identify a reference-stronger/target-weaker asymmetry, rather than literal correctness.

Table 8: Full IFEval results (prompt-level loose accuracy, %) for the proxy-based extension.

Model	Method	ar	bn	zh	fr	ja	es	Avg.
OLMoE	SFT	18.1 ±0.4	14.1 ±0.4	17.8 ±0.3	27.8 ±0.4	21.3 ±0.3	27.5 ±0.3	21.1 ±0.4
	RA-MoE	19.8 ±0.4	16.0 ±0.3	19.3 ±0.4	29.6 ±0.3	23.0 ±0.4	29.3 ±0.3	22.8 ±0.4
Qwen1.5	SFT	21.4 ±0.5	20.5 ±0.4	22.3 ±0.5	28.9 ±0.3	25.8 ±0.4	29.6 ±0.4	24.8 ±0.4
	RA-MoE	24.5 ±0.3	22.2 ±0.4	24.5 ±0.4	31.1 ±0.4	27.7 ±0.4	32.0 ±0.4	27.0 ±0.4
DSV2	SFT	18.2 ±0.4	15.3 ±0.3	24.6 ±0.4	30.5 ±0.3	26.4 ±0.3	30.0 ±0.2	24.2 ±0.3
	RA-MoE	20.6 ±0.4	17.7 ±0.4	26.9 ±0.4	32.7 ±0.3	28.6 ±0.4	32.3 ±0.3	26.5 ±0.4

IFEval translation and evaluation. Because the original IFEval verifier assumes English punctuation conventions, we adapt sentence-boundary and comma-based constraints to the six target languages. Sentence-ending punctuation is extended to the corresponding native conventions, and comma detection recognizes both Latin and script-specific commas. Translation-quality scores for both the OLMoE SFT Mix training data and the translated IFEval evaluation set are reported in Table 7.

Full IFEval results. The main paper reports results averaged over the six target languages. Table 8 provides the full 18 model–language comparisons underlying the finding that proxy-based RA-MoE outperforms SFT in every setting.

Human validation of *proxy-ci*. We manually inspect 600 randomly sampled *proxy-ci* examples, with 100 examples from each target language. Annotators proficient in both the target language and English assign each example to one of three categories: (i) valid *proxy-ci*, where the English response is acceptable and the target-language response is clearly weaker or fails; (ii) a false positive primarily caused by translation or formatting artifacts; or (iii) ambiguous. Under this protocol, 85.2% of examples match the intended English-stronger/target-weaker pattern, 7.1% are translation/formatting false positives, and 7.7% are ambiguous. These results support PPL-derived *proxy-ci* as a practical sample-selection signal despite its noise.

B Additional Method Details

B.1 Entropy-Normalized Jensen–Shannon Divergence

For two routing distributions $\mathbf{p}$ and $\mathbf{q}$ over E experts, we first add a small constant to every component and renormalize:

$$\bar{p}_e = \frac{p_e + \epsilon}{\sum_{e'}(p_{e'} + \epsilon)}, \qquad \bar{q}_e = \frac{q_e + \epsilon}{\sum_{e'}(q_{e'} + \epsilon)}, \qquad \epsilon = 10^{-12}. \tag{11}$$

All logarithms used in this diagnostic are base two. Let

$$\bar{\mathbf{m}} = \frac{\bar{\mathbf{p}} + \bar{\mathbf{q}}}{2}. \tag{12}$$

The Jensen–Shannon divergence is computed as

$$\mathrm{JS}_2(\bar{\mathbf{p}}, \bar{\mathbf{q}}) = \frac{1}{2} D_{\mathrm{KL},2}\left(\bar{\mathbf{p}} \,\|\, \bar{\mathbf{m}}\right) + \frac{1}{2} D_{\mathrm{KL},2}\left(\bar{\mathbf{q}} \,\|\, \bar{\mathbf{m}}\right), \tag{13}$$

where $D_{\mathrm{KL},2}$ denotes KL divergence with base-2 logarithms. We define the base-2 entropy as

$$H_2(\mathbf{p}) = -\sum_e p_e \log_2 p_e, \tag{14}$$

and the mean entropy of the two routing distributions as

$$\overline{H}_2(\bar{\mathbf{p}}, \bar{\mathbf{q}}) = \frac{1}{2}\left[H_2(\bar{\mathbf{p}}) + H_2(\bar{\mathbf{q}})\right]. \tag{15}$$

The entropy-normalized divergence reported in the paper is

$$D_{\mathrm{JS}}(\mathbf{p}, \mathbf{q}) = \frac{\mathrm{JS}_2(\bar{\mathbf{p}}, \bar{\mathbf{q}})}{\log_2 E - \overline{H}_2(\bar{\mathbf{p}}, \bar{\mathbf{q}})}. \tag{16}$$

The denominator measures the gap between the maximum entropy attainable over E experts and the average entropy of the two routing distributions. If this denominator is at most 10^{-10}, the implementation returns zero. No additional clipping is applied.

Table 9: Selected middle-layer ranges used in the core experiments. All ranges are inclusive, zero-based absolute transformer-layer indices.

Model	Task	ar	bn	zh	fr	ja	es
OLMoE	GSM8K	5–11	4–9	5–11	4–11	4–10	4–11
	LogicBench	4–11	4–11	4–11	4–11	4–11	4–11
	MMLU	6–12	6–13	6–13	5–12	4–11	4–11
Qwen1.5	GSM8K	5–14	5–15	5–13	5–15	5–13	4–13
	LogicBench	5–11	5–11	4–12	4–11	4–15	4–12
	MMLU	8–16	8–16	8–16	6–16	8–16	7–16
DSV2	GSM8K	6–16	6–14	7–15	6–15	6–16	7–14
	LogicBench	9–16	11–16	9–16	10–15	9–16	8–15
	MMLU	10–17	9–16	9–15	11–16	9–15	10–16

B.2 Middle-Layer Identification

The middle-layer interval is identified independently for each model–task–target-language configuration. For a fixed configuration, we first average the per-example routing divergence at every MoE layer:

$$\mathrm{Div}^{(\ell)} = \frac{1}{N}\sum_{i=1}^{N} D_{i,\ell}, \tag{17}$$

where

$$D_{i,\ell} = D_{\mathrm{JS}}\left(\mathbf{q}^{(\mathrm{en},\ell,i)}, \mathbf{q}^{(\mathrm{tgt},\ell,i)}\right) \tag{18}$$

is the entropy-normalized cross-lingual routing divergence for example i at layer ℓ.

We then set the divergence threshold to the median of the layer-wise means:

$$\tau = \mathrm{median}_{\ell}\left\{\mathrm{Div}^{(\ell)}\right\}. \tag{19}$$

Layers satisfying

$$\mathrm{Div}^{(\ell)} \le \tau \tag{20}$$

are retained, and the selected middle-layer range $\mathcal{L}_{\mathrm{mid}}$ is defined as the longest contiguous interval among these layers. If multiple intervals have the same maximum length, the earliest interval in layer order is retained. Consequently, $\mathcal{L}_{\mathrm{mid}}$ is determined separately for every model–task–target-language configuration and is not shared across target languages.

Table 9 reports the selected ranges used in the core experiments. All ranges are inclusive, zero-based absolute transformer-layer indices.

B.3 Derivation of the Routing Alignment Decomposition

We provide the derivation of the routing-alignment decomposition used in Section 3. For clarity, we omit the sample and layer indices (i, ℓ) throughout this section.

Let $\mathcal{E}^{\mathrm{task}}$ denote the task-expert set and define the total routing mass assigned to these experts as

$$m^{(\mathrm{lang})} = \sum_{e\in\mathcal{E}^{\mathrm{task}}} q_e^{(\mathrm{lang})}. \tag{21}$$

The normalized routing distribution within the task-expert set is

$$\tilde{q}_e^{(\mathrm{lang})} = \frac{q_e^{(\mathrm{lang})}}{m^{(\mathrm{lang})}}, \qquad e \in \mathcal{E}^{\mathrm{task}}, \tag{22}$$

such that

$$q_e^{(\mathrm{lang})} = m^{(\mathrm{lang})}\tilde{q}_e^{(\mathrm{lang})}, \qquad \sum_{e\in\mathcal{E}^{\mathrm{task}}} \tilde{q}_e^{(\mathrm{lang})} = 1. \tag{23}$$

Recall that RA-MoE constructs the collapsed routing distribution

$$\mathbf{r}^{(\text{lang})} = \left[\left(q_e^{(\text{lang})} \right)_{e \in \mathcal{E}^{\text{task}}},\ 1 - m^{(\text{lang})} \right], \tag{24}$$

which preserves the routing probability assigned to each identified task expert while aggregating all remaining experts into a single residual component.

The KL divergence between the English and target-language collapsed distributions can be expanded as

$$\begin{aligned} & D_{\mathrm{KL}} \left(\mathbf{r}^{(\mathrm{en})} \,\middle\|\, \mathbf{r}^{(\mathrm{tgt})} \right) \\ &= \sum_{e \in \mathcal{E}^{\text{task}}} q_e^{(\mathrm{en})} \log \frac{q_e^{(\mathrm{en})}}{q_e^{(\mathrm{tgt})}} + \left(1 - m^{(\mathrm{en})} \right) \log \frac{1 - m^{(\mathrm{en})}}{1 - m^{(\mathrm{tgt})}}. \end{aligned} \tag{25}$$

Substituting

$$q_e^{(\text{lang})} = m^{(\text{lang})} \tilde{q}_e^{(\text{lang})} \tag{26}$$

into the first term gives

$$\begin{aligned} & \sum_{e \in \mathcal{E}^{\text{task}}} m^{(\mathrm{en})} \tilde{q}_e^{(\mathrm{en})} \log \frac{m^{(\mathrm{en})} \tilde{q}_e^{(\mathrm{en})}}{m^{(\mathrm{tgt})} \tilde{q}_e^{(\mathrm{tgt})}} \\ &= m^{(\mathrm{en})} \sum_{e \in \mathcal{E}^{\text{task}}} \tilde{q}_e^{(\mathrm{en})} \left[\log \frac{m^{(\mathrm{en})}}{m^{(\mathrm{tgt})}} + \log \frac{\tilde{q}_e^{(\mathrm{en})}}{\tilde{q}_e^{(\mathrm{tgt})}} \right] \\ &= m^{(\mathrm{en})} \log \frac{m^{(\mathrm{en})}}{m^{(\mathrm{tgt})}} + m^{(\mathrm{en})} D_{\mathrm{KL}} \left(\tilde{\mathbf{q}}^{(\mathrm{en})} \,\middle\|\, \tilde{\mathbf{q}}^{(\mathrm{tgt})} \right), \end{aligned} \tag{27}$$

where the final equality follows from

$$\sum_{e \in \mathcal{E}^{\text{task}}} \tilde{q}_e^{(\mathrm{en})} = 1. \tag{28}$$

We next define the binary distribution describing routing mass inside and outside the task-expert set:

$$\mathbf{b}^{(\text{lang})} = \left[m^{(\text{lang})}, 1 - m^{(\text{lang})} \right]. \tag{29}$$

Its KL divergence is

$$\begin{aligned} D_{\mathrm{KL}} \left(\mathbf{b}^{(\mathrm{en})} \,\middle\|\, \mathbf{b}^{(\mathrm{tgt})} \right) = m^{(\mathrm{en})} \log \frac{m^{(\mathrm{en})}}{m^{(\mathrm{tgt})}} \\ + \left(1 - m^{(\mathrm{en})} \right) \log \frac{1 - m^{(\mathrm{en})}}{1 - m^{(\mathrm{tgt})}}. \end{aligned} \tag{30}$$

Combining Equations 25, 27, and 30 yields

$$D_{\mathrm{KL}} \left(\mathbf{r}^{(\mathrm{en})} \,\middle\|\, \mathbf{r}^{(\mathrm{tgt})} \right) = D_{\mathrm{KL}} \left(\mathbf{b}^{(\mathrm{en})} \,\middle\|\, \mathbf{b}^{(\mathrm{tgt})} \right) + m^{(\mathrm{en})} D_{\mathrm{KL}} \left(\tilde{\mathbf{q}}^{(\mathrm{en})} \,\middle\|\, \tilde{\mathbf{q}}^{(\mathrm{tgt})} \right). \tag{31}$$

The first term measures the discrepancy in the total routing mass assigned to the task-expert set, whereas the second measures the discrepancy in the normalized allocation within that set. The factor $m^{(\mathrm{en})}$ follows directly from the KL decomposition and weights within-set allocation differences more strongly when the successful English routing assigns greater probability mass to the identified task experts.

C Experimental Implementation

C.1 Model Details

We evaluate RA-MoE on three pretrained MoE models spanning different architectures and parameter scales. Table 10 summarizes their key architectural properties.

We use the following model variants in all main experiments: `allenai/OLMoE-1B-7B-0924-SFT`, `Qwen/Qwen1.5-MoE-A2.7B`, and `deepseek-ai/DeepSeek-V2-Lite-Chat`.

Table 10: Architectural summary of the three MoE models. †Qwen1.5-MoE and DS-V2-Lite include four and two shared experts activated for every token in addition to the routed experts, respectively. DS-V2-Lite has one dense layer followed by 26 MoE layers.

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+4†	4+4†
DS-V2-Lite	16B	2.4B	27	26	64+2†	6+2†

Table 11: Hyperparameter settings shared across the core 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	2×10^{-5}
LR scheduler	Linear
Warmup ratio	0.03
Weight decay	0.0
Gradient clipping	None
Epochs	1
Precision	bfloat16
RA-MoE specific	
Alignment weight λ	1.0
Task experts per layer K	8
Middle-layer threshold τ	median (50th percentile)
General corpus (Stage 2)	FLORES-200
Routing-profiling batch size	32

C.2 Training Hyperparameters

All training-based methods use LoRA (Hu et al., 2022) on the up and down projections of each FFN/MoE expert layer, with the remaining model parameters frozen. SFT, RISE, and RA-MoE use the same optimization schedule and data order within each comparison.

Table 12 records settings that depend on the model/task pair.

C.3 Routing-Alignment Implementation

For a minibatch $\mathcal{B}$, let $\mathcal{B}_{\text{ci}} \subseteq \mathcal{B}$ denote the examples selected for routing alignment. The implementation averages the KL divergence over the selected examples and middle layers:

$$\widehat{\mathcal{L}}_{\text{align}} = \frac{1}{|\mathcal{B}_{\text{ci}}|} \sum_{i \in \mathcal{B}_{\text{ci}}} \frac{1}{|\mathcal{L}_{\text{mid}}|} \sum_{\ell \in \mathcal{L}_{\text{mid}}} D_{\text{KL}}\Big(\mathbf{r}^{(\text{en},\ell,i)} \,\Big\|\, \mathbf{r}^{(\text{tgt},\ell,i)}\Big). \tag{32}$$

When a local minibatch contains no selected examples, we set $\widehat{\mathcal{L}}_{\text{align}} = 0$. The English routing reference is fixed, while the target-language routing distribution is computed from the current SFT forward pass. The final objective is $\mathcal{L}_{\text{CE}} + \lambda \widehat{\mathcal{L}}_{\text{align}}$; all examples contribute to $\mathcal{L}_{\text{CE}}$, whereas only selected *ci* (or *proxy-ci* in the IFEval extension) examples contribute to the alignment term.

Order of aggregation. For each sample and selected middle layer, we first average router probabilities over assistant-response tokens to obtain a sequence-level routing distribution. We then construct

Table 12: Per-model, per-task batch-size and sequence-length settings for the core experiments. Effective batch size includes gradient accumulation across four GPUs.

		GSM8K	LogicBench	MMLU
OLMoE	Micro-batch size	16	8	8
	Gradient accum. steps	4	8	8
	Effective batch size	256	256	256
	Max sequence length	512	2048	2048
DSV2	Micro-batch size	4	3	3
	Gradient accum. steps	4	2	2
	Effective batch size	64	24	24
	Max sequence length	512	2048	2048
Qwen1.5	Micro-batch size	4	3	3
	Gradient accum. steps	4	2	2
	Effective batch size	64	24	24
	Max sequence length	512	2048	2048

the collapsed routing distribution $\mathbf{r}$ and compute the KL divergence between the English and target-language distributions. The resulting KL values are averaged across selected middle layers and *ci* examples in the minibatch.

Distributed training. For each local micro-batch, each worker identifies its local selected examples and averages the per-example alignment losses over this subset. If no selected examples are present, the local alignment loss is set to zero. Standard distributed data parallelism then averages gradients equally across workers, without reweighting by the number of selected examples on each worker. Consequently, the resulting routing gradient corresponds to an average of worker-local means rather than an exact mean over all selected examples in the global batch. With gradient accumulation, this computation is performed independently for each micro-batch.

C.4 Computational Infrastructure

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

C.5 Evaluation Settings

The benchmark splits, metrics, and few-shot settings are summarized in Table 6. All evaluations use vLLM. GSM8K and LogicBench use greedy decoding, with answers scored using the task-specific procedures described in Section A.3. MMLU is evaluated 0-shot using constrained multiple-choice scoring over $\{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$, selecting the option with the highest next-token log probability.

For each model–method setting, we repeat evaluation five times using random seeds $\{42, 123, 456, 789, 2026\}$ and report the mean accuracy and standard deviation across these runs.

C.6 Baseline Reimplementation Details

Routing Steering (RS). Routing Steering (Bandarkar et al., 2026) is applied at inference time without updating model parameters. We follow the original implementation and default settings, using the same middle-layer region as RA-MoE and without task- or language-specific hyperparameter tuning.

RISE. RISE (Zheng et al., 2026) is reproduced following the original implementation and default expert-selection procedure. The selected language-specific expert subnetwork is adapted using the same training data, optimizer, learning rate, and training schedule as RA-MoE, without task- or language-specific hyperparameter tuning.

Table 13: Pairwise routing consistency across successful reference languages on DeepSeek-V2-Lite-Chat (GSM8K).

Pair	Jaccard	Mass diff.	Within-set JS
en–zh	0.4182	0.00923	0.00281
en–es	0.4096	0.00844	0.00226
zh–es	0.5467	0.01022	0.00333

D Additional Analysis Details

D.1 Cross-Language Transfer Experiments

Reference-language transfer. Section 5.3 evaluates whether successful routing supervision must come from English. Both experiments target Japanese on GSM8K. For DeepSeek-V2-Lite-Chat, the reference language is English, Chinese, or Spanish; for Qwen1.5-MoE, the reference is English, French, or Spanish. Each non-English reference is first adapted with standard SFT to establish sufficient task competence. We then identify its task experts analogously to the English-reference setting and use successful routing behavior in that language as the alignment reference. For each reference–target pair, the correctness groups are recomputed independently, so *ci* denotes examples on which the current reference language is correct and Japanese is incorrect.

Target-side task-expert transfer. For RA-MoE†, we identify the middle-layer range, per-layer task-expert identities, and per-example English reference routing using the Chinese target-language configuration for each model–task pair. These components are then reused unchanged when fine-tuning Arabic or Bengali, while all other training settings remain the same. Thus, RA-MoE† directly tests whether a task-routing configuration identified for one target language can be transferred to another without re-running the identification or reference-routing stages.

D.2 Mechanistic Analysis of Language-Shared Task Routing

We compare GSM8K task routing across English, Chinese, and Spanish on DeepSeek-V2-Lite-Chat. For routing-pattern comparisons, we use a shared set of 12,388 parallel examples that are correctly solved in all three reference languages but incorrectly solved in Japanese. Specifically, this set is the intersection of the pair-specific English–Japanese, Chinese–Japanese, and Spanish–Japanese *ci* subsets. This controls sample composition across reference languages.

Task-expert identity. For task-expert sets $\mathcal{E}_A^{(\ell)}$ and $\mathcal{E}_B^{(\ell)}$, we compute layer-wise Jaccard similarity

$$J^{(\ell)}(A,B) = \frac{|\mathcal{E}_A^{(\ell)} \cap \mathcal{E}_B^{(\ell)}|}{|\mathcal{E}_A^{(\ell)} \cup \mathcal{E}_B^{(\ell)}|}, \tag{33}$$

and average over the selected middle layers. The observed pairwise cross-language similarities are 0.4182 (en–zh), 0.4096 (en–es), and 0.5467 (zh–es), yielding a mean of approximately 0.458. As a cross-task control, English GSM8K vs. English MMLU yields a mean Jaccard similarity of 0.0543.

Random-expert control. For $E = 64$ experts and $K = 8$ selected experts per layer, we draw 10^6 pairs of random size-K subsets and compute their Jaccard similarity. The random baseline has mean Jaccard 0.0706. The empirical probability of obtaining a random Jaccard at least as large as the smallest observed cross-language value (0.4096) is 3.57×10^{-4}.

Task-expert usage. We additionally compare the total routing mass assigned to task experts and the normalized routing distribution within the shared task-expert set. The mean task-expert masses are 0.1837 (en), 0.1826 (zh), and 0.1752 (es). Pairwise mean absolute mass differences are 0.00923 (en–zh), 0.00844 (en–es), and 0.01022 (zh–es). The corresponding within-set JS divergences are 0.00281, 0.00226, and 0.00333.

Table 14: Wall-clock time (hours) decomposed by stage. S1+2 denotes the two preparatory stages and S3 denotes fine-tuning. *Per-lang.* runs the routing-identification pipeline separately for each target-language configuration, whereas *reuse* transfers the identified middle-layer range, task experts, and English reference routing across target languages.

		GSM8K		**LogicBench**		**MMLU**	
Model	**Method**	**S1+2**	**S3**	**S1+2**	**S3**	**S1+2**	**S3**
OLMoE	SFT	–	3.0	–	1.2	–	3.5
	RA-MoE (per-lang.)	0.5	3.0	0.2	1.2	0.5	3.5
	RA-MoE (reuse)	0.3	3.0	0.1	1.2	0.3	3.5
Qwen1.5	SFT	–	15.5	–	1.5	–	18.5
	RA-MoE (per-lang.)	2.0	15.0	0.2	1.5	3.0	18.5
	RA-MoE (reuse)	1.0	15.0	0.1	1.5	1.5	18.5
DSV2	SFT	–	12.0	–	1.5	–	18.0
	RA-MoE (per-lang.)	1.5	11.5	0.2	1.5	3.0	18.0
	RA-MoE (reuse)	0.8	11.5	0.1	1.5	1.5	18.0

E Computational Cost Analysis

We analyze RA-MoE overhead in terms of preparatory inference time, training-time memory, and additional routing-alignment arithmetic. Stage 1 uses autoregressive inference to categorize training examples and Stage 2 uses teacher-forced forward passes to profile routing. Neither stage requires gradient computation. Stage 3 uses the same model forward pass as standard SFT and adds the routing-alignment loss to already available router outputs; it does not require an additional model forward/backward pass.

E.1 Training Time Analysis

RA-MoE introduces two preparatory stages before fine-tuning: Stage 1 constructs the correctness-based sample groups, and Stage 2 profiles routing behavior to identify the selected middle layers and task experts and to obtain the reference routing signals. Both stages require only inference, without gradient computation. During Stage 3, the routing alignment loss is computed from router probabilities already produced by the SFT forward pass and therefore does not require an additional model forward or backward pass.

Table 14 reports wall-clock time for all three models and core tasks. Stage 3 training time remains comparable to standard SFT across all settings. When the preparatory stages are run separately for each target-language configuration, the Stage 1+2 cost is at most 16.7% of Stage 3. Reusing the middle-layer range, per-layer task-expert identities, and English reference routing across target languages further reduces this overhead to at most 10.0%.

E.2 Memory Overhead

RA-MoE introduces no additional trainable model parameters beyond the LoRA adapters used by standard SFT. Under the matched OLMoE configuration, both SFT and RA-MoE contain 100,663,296 trainable parameters. The stored English reference routing distributions are kept in CPU memory as float32 tensors and transferred to the GPU only for the selected examples in the current batch.

During Stage 3, RA-MoE additionally retains router outputs from the selected middle layers to compute the alignment loss. In the available training logs, SFT and RA-MoE show the same allocated GPU memory after optimizer initialization (13.26 GB in the matched OLMoE configuration). Thus, RA-MoE does not introduce additional model, LoRA, or optimizer states; its extra runtime memory is limited to the routing references and intermediate router outputs required by the alignment objective.

E.3 ALIGNMENT-LOSS COMPUTATION

Once the router probabilities are available from the standard forward pass, RA-MoE computes the alignment loss only for *ci* examples and selected middle layers. Since routing probabilities are first averaged over response tokens, the KL divergence is computed once per selected example and layer rather than once per token. The collapsed distribution contains only $K + 1$ entries—the K task experts and one residual component—so, following the same coarse arithmetic accounting used for the base-model computation, the alignment cost scales as

$$\mathrm{FLOPs}_{\mathrm{align}} \approx 2|\mathcal{B}_{\mathrm{ci}}||\mathcal{L}_{\mathrm{mid}}|(K+1). \tag{34}$$

To provide a conservative numerical comparison, we use OLMoE with the same representative configuration as the base-computation estimate: $B = 4096$ tokens, $L = 16$ MoE layers, $k_{\mathrm{act}} = 8$ activated experts per token, $d_m = 2048$, $d_e = 1024$, and LoRA rank $r = 16$. The expert FFN and LoRA computations are

$$\mathrm{FLOPs}_{\mathrm{base}} = 6BLk_{\mathrm{act}}d_m d_e = 6597.1 \text{ GFLOPs}, \tag{35}$$

$$\mathrm{FLOPs}_{\mathrm{LoRA}} = 4BLk_{\mathrm{act}}r(d_m + d_e) = 103.1 \text{ GFLOPs}. \tag{36}$$

For RA-MoE, $K = 8$. Even under the deliberately conservative assumption that all $B = 4096$ token positions correspond to distinct selected examples and that alignment is applied to all $L = 16$ MoE layers, the alignment computation is bounded by

$$\mathrm{FLOPs}_{\mathrm{align}} \leq 2BL(K+1) \approx 0.00118 \text{ GFLOPs}. \tag{37}$$

This upper bound is only about $1.8 \times 10^{-5}\%$ of the approximately 6700.1 GFLOPs required by the expert and LoRA computations. The actual overhead is smaller because alignment is applied only to *ci* examples and selected middle layers. Thus, the arithmetic cost of the routing-alignment objective is negligible relative to the expert and LoRA computations of the underlying MoE model.