Decoding Knowledge Attribution in Mixture-of-Experts: A Framework of Basic-Refinement Collaboration and Efficiency Analysis

Junzhuo Li[12], **Bo Wang**[1], **Xiuze Zhou**[12], **Peijie Jiang**[3], **Jia Liu**[3] and **Xuming Hu**[12]*
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2] The Hong Kong University of Science and Technology
[3]Ant Group
{jz.li, bo.wang, xz.zhou}@connect.hkust-gz.edu.cn
xuminghu@hkust-gz.edu.cn

Abstract

The interpretability of Mixture-of-Experts (MoE) models, especially those with heterogeneous designs, remains underexplored. Existing attribution methods for dense models fail to capture dynamic routing-expert interactions in sparse MoE architectures. To address this issue, we propose a cross-level attribution algorithm to analyze sparse MoE architectures (Qwen 1.5-MoE, OLMoE, Mixtral-8x7B) against dense models (Qwen 1.5-7B, Llama-7B, Mistral-7B). Results show MoE models achieve 31% higher per-layer efficiency via a "mid-activation, late-amplification" pattern: early layers screen experts, while late layers refine knowledge collaboratively. Ablation studies reveal a "basic-refinement" framework—shared experts handle general tasks (entity recognition), while routed experts specialize in domain-specific processing (geographic attributes). Semantic-driven routing is evidenced by strong correlations between attention heads and experts ($r = 0.68$), enabling task-aware coordination. Notably, architectural depth dictates robustness: deep Qwen 1.5-MoE mitigates expert failures (e.g., 43% MRR drop in geographic tasks when blocking top-10 experts) through shared expert redundancy, whereas shallow OLMoE suffers severe degradation (76% drop). Task sensitivity further guides design: core-sensitive tasks (geography) require concentrated expertise, while distributed-tolerant tasks (object attributes) leverage broader participation. These insights advance MoE interpretability, offering principles to balance efficiency, specialization, and robustness.

1 Introduction

In recent years, transformer-based large language models (LLMs) (Vaswani, 2017; Brown et al., 2020; Touvron et al., 2023b; Bai et al., 2023) have significantly enhanced task performance by scaling up parameters, yet at the cost of prohibitive computational demands. To balance efficiency and performance, Mixture-of-Experts (MoE) architectures (Fedus et al., 2021; Lepikhin et al., 2021; Shen et al., 2023; Jiang et al., 2024; Dai et al., 2024; Sun et al., 2024) activate subsets of experts per input, reducing overhead while maintaining capability. While existing research has prioritized optimizing routing mechanisms and expert module efficiency (e.g., top-k gating (Fedus et al., 2021; Lepikhin et al., 2021)), a critical question remains unanswered: *How do experts collaboratively process and refine knowledge within MoE models?*

This gap in interpretability is particularly pronounced in models with heterogeneous designs, such as shared expert modules (Dai et al., 2024). Despite their widespread adoption (DeepSeek-AI et al., 2024; Team, 2024), the roles of shared experts—whether they serve as universal feature extractors or redundant backups—lack empirical validation. The opacity of expert collaboration not only hampers credibility assessment (e.g., verifying if experts specialize in specific knowledge domains) but also forces model designers to rely on trial-and-error strategies, limiting systematic improvements. For instance, fundamental issues like the distribution of shared experts across layers and their impact on knowledge consistency remain unexplored.

Two key challenges compound this problem: (1) Existing attribution methods (Geva et al., 2021; Dai et al., 2022a; Geva et al., 2022; Wu et al., 2023) are designed for dense models and fail to capture dynamic interactions between routed experts and shared modules. (2) Comparative studies on heterogeneous MoE architectures are scarce, leaving functional assumptions (e.g., "shared experts handle general tasks") unverified.

In this paper, we address these gaps by proposing a cross-level attribution algorithm for MoE models, extending dense model methods to ana-

*Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22431–22446

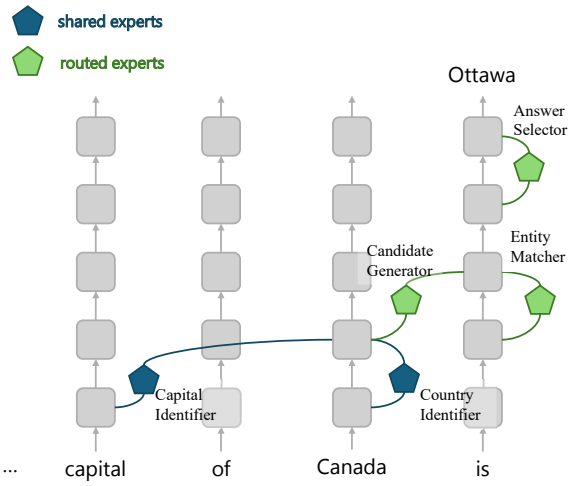

Figure 1: Schematic of the MoE expert layer: Shared experts perform foundational tasks (e.g., the Capital Identifier recognizes "capital," the Country Identifier recognizes "Canada"), while routed experts execute refinement steps—generating candidate capitals, matching them to "Canada," and selecting the final answer ("Ottawa").

lyze both macro architectures (Mixtral-8x7B (Jiang et al., 2024), Qwen 1.5-MoE (Team, 2024), OLMoE (Muennighoff et al., 2024)) and micro behaviors. Our experiments uncover four pivotal insights. First, MoE models significantly outperform dense counterparts in per-layer efficiency—Qwen 1.5-MoE achieves a layer efficiency of 0.294 with 24 layers, surpassing Qwen 1.5-7B (0.213 with 32 layers), a 37% improvement attributed to concentrated knowledge processing in early layers. Second, we identify a hierarchical "basic-refinement" collaboration: shared experts act as generalists for foundational tasks like entity localization, while routed experts specialize in refining domain-specific attributes (e.g., linking "Canada" to "Ottawa"). Third, semantic-driven routing is validated through strong temporal correlations between attention heads and expert selection ($r = 0.68, p < 0.0014$), indicating that attention mechanisms guide experts toward task-relevant features. Finally, robustness varies starkly by architecture: blocking top-10 experts in deep Qwen 1.5-MoE causes a 43% MRR drop in geographic tasks, whereas shallow OLMoE suffers a 76% decline, underscoring the critical role of depth and shared experts in redundancy design.

Contributions Our main contributions are as follows:

- **A cross-level attribution algorithm for MoE models**, bridging the gap in interpreting sparse architectures by dynamically weighting expert and attention contributions.
- **Empirical validation of MoE efficiency and collaboration patterns**, including layer efficiency gains, a "mid-activation, late-amplification" processing paradigm, and semantic-driven expert routing.
- **Design implications for robust MoE architectures**, demonstrating that deep models with shared experts balance specialization and redundancy, particularly critical for complex tasks like geographic reasoning.

2 Methodology

In this section, we first describe the MoE model architecture, followed by the knowledge attribution method. More details for model settings and experimental settings (i.e., dataset, evaluation metrics) can be found in Appendix A.

2.1 MoE Architecture

In the decoder-only MoE architecture, each Transformer decoder block replaces the conventional Feed-Forward Network (FFN) with an MoE layer, enhancing model capacity via dynamic routing. Given an input sequence $X = (x_1, x_2, \ldots, x_T)$, the tokens are embedded to produce the initial hidden state $\boldsymbol{h}^0 = \text{Embed}(X)$. We omit bias and layer normalization (Ba et al., 2016) following Geva et al. (2022).

Each decoder layer consists of a self-attention mechanism and an MoE layer. At layer l, the attention output for token x_i is computed as:

$$\boldsymbol{A}_i^l = \sum_{j=1}^{H} \text{ATTN}_j^l(\boldsymbol{h}_1^{l-1}, \boldsymbol{h}_2^{l-1}, \ldots, \boldsymbol{h}_T^{l-1}), \quad (1)$$

where H is the number of attention heads. The intermediate representation $\boldsymbol{u}_i^l = \boldsymbol{A}_i^l + \boldsymbol{h}_i^{l-1}$ is routed to N experts via a gating network, which computes the routing probabilities:

$$\boldsymbol{g}_i^l = \text{softmax}(\boldsymbol{W}_g^l \boldsymbol{u}_i^l + \boldsymbol{b}_g^l), \quad (2)$$

where $\boldsymbol{W}_g^l$ and $\boldsymbol{b}_g^l$ are learnable parameters. Each expert $\mathcal{E}_j^l$ is a feed-forward network:

$$\mathcal{E}_j^l(\boldsymbol{u}_i^l) = \boldsymbol{W}_{j,2}^l \phi(\boldsymbol{W}_{j,1}^l \boldsymbol{u}_i^l + \boldsymbol{b}_j^l), \quad (3)$$

Model	HIT@10	MRR	Total FFN Gain	Total Attn Gain	Peak Gain Layer Position (Rel. Position%)	Layer Efficiency (FFN Gain / Layers)
Llama-7B	0.90	0.70	5.16	4.03	24.82 (77.6%)	0.161
Qwen 1.5-7B	0.79	0.62	6.49	4.14	27.08 (90.2%)	0.203
Mistral-7B	0.88	0.71	6.74	2.96	26.64 (83.2%)	0.211
Qwen 1.5-MoE	0.85	0.63	7.36	3.18	20.36 (84.8%)	0.307
OLMoE	0.83	0.64	4.98	4.40	13.53 (84.6%)	0.311
Mixtral-8x7B	0.90	0.73	6.79	3.03	26.66(83.3%)	0.212

Table 1: Comparison of model architectures on performance metrics, component contributions, and layer efficiency across dense and MoE models. Rel.:Relative.

where $\phi(\cdot)$ is a nonlinear activation function (Nair and Hinton, 2010; Maas et al., 2013; Hendrycks and Gimpel, 2016). The final output for token x_i at layer l is computed by a weighted sum of the top k experts' outputs:

$$\boldsymbol{F}_i^l = \sum_{j=1}^{k} g_{i,j}^l \cdot \mathcal{E}_j^l(\boldsymbol{u}_i^l). \tag{4}$$

Some MoEs include shared experts that are always selected (Dai et al., 2024), which results in:

$$\boldsymbol{F}_i^l = \sum_{j=1}^{k} g_{i,j}^l \cdot \mathcal{E}_j^l(\boldsymbol{u}_i^l) + g_{i,s}^l \cdot \mathcal{E}_s^l(\boldsymbol{u}_i^l). \tag{5}$$

Finally, the output at position i of the l-th layer is:

$$\boldsymbol{h}_i^l = \boldsymbol{h}_i^{l-1} + \boldsymbol{A}_i^l + \boldsymbol{F}_i^l. \tag{6}$$

The final hidden states $\boldsymbol{h}_i^L$ need to go through the unembedding process to map the high-dimensional representations back to the vocabulary space. Specifically, for each position i, the final output token y_i is obtained by multiplying the hidden state $\boldsymbol{h}_i^L$ with the unembedding matrix $\boldsymbol{W}_{\text{unembed}}$ and passing it through a softmax function.

2.2 Neuron-Level Attribution

We adopt the neuron-level attribution methods from Yu and Ananiadou (2024) for the FFN and attention layers and extend them for MoE models. For attention neurons $\boldsymbol{v}_A^l$, a vector (neuron) in l-th Attention layer, the importance score is:

$$\mathcal{I}(\boldsymbol{v}_A^l) = \log p(y_i|\boldsymbol{v}_A^l + \boldsymbol{h}^{l-1}) - \log p(y_i|\boldsymbol{h}^{l-1}), \tag{7}$$

For a token x_i The importance score for FFN neurons is defined as the log probability increase:

$$\mathcal{I}(\boldsymbol{v}_F^l) = \log p(y_i|\boldsymbol{v}_F^l + \boldsymbol{u}^l) - \log p(y_i|\boldsymbol{u}^l), \tag{8}$$

where $\boldsymbol{v}_F^l$ is a vector (neuron) in l-th FFN layer.

In MoE models, we extend these methods to account for dynamic routing and expert contributions. The importance score for MoE expert neurons is computed as:

$$\mathcal{I}(\boldsymbol{v}_{\mathcal{E}_j}^l) = \log p(y_i|g_{i,j}^l \boldsymbol{v}_{\mathcal{E}_j}^l + \boldsymbol{u}^l) - \log p(y_i|\boldsymbol{u}^l), \tag{9}$$

where $g_{i,j}^l$ is the gating probability for expert $\mathcal{E}_j$, and $v_{\mathcal{E}_j}^l$ is the neuron from expert $\mathcal{E}_j$ of l-th layer.

Through these modifications, we are able to accurately attribute importance not only to the conventional FFN and attention neurons but also to the individual experts and their specific contributions within the MoE framework. This extension allows us to capture the complex interactions between experts and tokens, providing deeper insights into the behavior of MoE models and how they utilize expert knowledge.

The importance score represents the influence of a neuron on the final prediction, reflecting the extent of its contribution during the model's inference process. More specifically, it can be regarded as the gain introduced by the module to the overall model performance, revealing the role of different neurons or experts in the model's output. This metric aids in understanding how the model utilizes the knowledge of various components and the significance of each module for specific tasks.

3 Comparative Analysis: Efficiency and Knowledge Processing in MoE vs. Dense Models

In this section, we compare three MoE architectures (Qwen 1.5-MoE, Mixtral, and OLMoE) with two dense models (Qwen 1.5-7B, Llama-7B, and

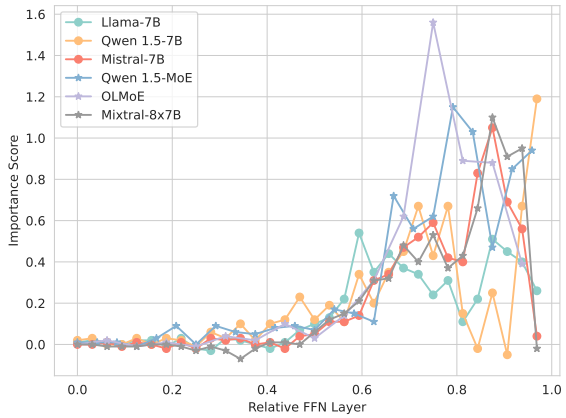

Figure 2: Comparison of layer importance scores distribution in FFN layers.

Mistral) across several key metrics. Our analysis highlights significant differences in parameter efficiency, knowledge localization, and FFN gain evolution—differences that underscore the unique interpretability advantages of MoE models.

3.1 Layer Efficiency and Knowledge Concentration in MoE Models

Table 1 summarizes the performance of the four models, revealing that MoE architectures achieve notably higher per-layer gains. For example, Qwen 1.5-MoE attains a layer efficiency of 0.307 using only 24 layers, whereas its dense counterpart, Qwen 1.5-7B, requires 32 layers to reach an efficiency of just 0.203. Despite a slightly lower overall FFN gain, OLMoE—by virtue of its shallower design—achieves the highest layer efficiency at 0.311, demonstrating a marked optimization in computational density.

The data further indicate that MoE models concentrate their knowledge processing at earlier stages. Specifically, Qwen 1.5-MoE and OLMoE reach their peak FFN gains at 84.8% and 84.6% of the network depth, respectively. This "early routing, late refinement" pattern enables these models to perform initial expert filtering in mid-layers and then focus on intensive knowledge refinement in later layers. In contrast, dense models such as Qwen 1.5-7B and Llama-7B display a more gradual accumulation of gains—peaking at 90.2% and 77.6% of the network depth—which, while beneficial for cross-layer integration, tends to blur the functional distinctions among layers and reduce interpretability.

Furthermore, a comparison of attention versus FFN gains underscores divergent processing strategies. Qwen 1.5-MoE, for instance, exhibits a pronounced reliance on specialized expert networks, with an FFN gain of 7.36 contrasted against an attention gain of only 3.18. Dense models, by comparison, distribute gains more evenly (with attention gains of 4.14 and 4.03 for Qwen 1.5-7B and Llama-7B, respectively), complicating the attribution of specific processing roles to individual layers. Notably, OLMoE achieves nearly identical attention and FFN gains (4.98 and 4.40, respectively), reflecting a balanced strategy that integrates both attention mechanisms and expert routing.

3.2 FFN Gain Evolution: Mid-Stage Activation and Late-Stage Amplification

Figure 2 illustrates the evolution of FFN gains across layers in Llama-7B, Qwen 1.5-MoE, Qwen 1.5-7B , OLMoE, Mistral, and Mixtral. The analysis reveals distinct processing patterns shaped by architectural designs:

Qwen 1.5-MoE exhibits a three-phase dynamic: (1) Early stage (Layers 1–13): Parallel initialization of experts extracts basic features (e.g., entity localization), contributing 6.1% of the total gain. (2) Mid-stage (Layers 14–19): Dynamic routing activates expert screening, where the top-4 gating mechanism selects task-relevant specialists. This phase contributes 43.5% of the total gain while minimizing redundant computations. (3) Late stage (Layers 20–24): Collaborative refinement between shared experts (e.g., broad knowledge integration) and routed experts (e.g., attribute association) amplifies gains, accounting for 50.4% of the total.

In contrast, Qwen 1.5-7B relies on linear accumulation across deeper layers (25–32), gradually integrating knowledge without explicit stage specialization. OLMoE, as a shallow architecture, rapidly peaks at layer 12 (mid-stage) but lacks sustained late-stage refinement due to limited depth, leading to a gradual performance decay.

Notably, Mistral diverges significantly from typical dense models: While other dense architectures exhibit diminished mid-to-late-stage gains, Mistral sustains high FFN activation through later layers, achieving gain levels comparable to MoE models. Mixtral shows strong similarity to Mistral, with both models peaking sharply at layer 28 before experiencing rapid performance decay. This parallel trajectory—absent in other architectures—suggests potential initialization-driven dynamics influencing their gain structures.

This structured processing enables MoE models

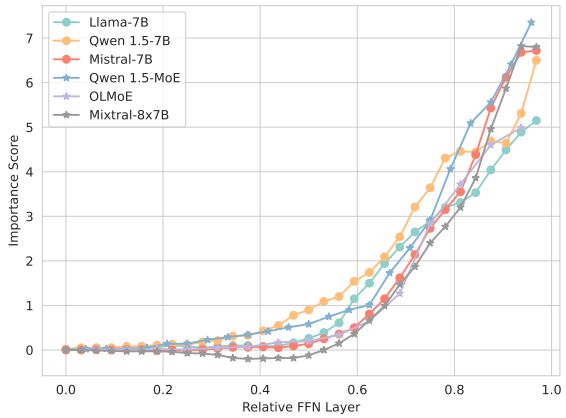

Figure 3: Cumulative FFN layers contribution curves.

to achieve higher per-layer efficiency. For example, Qwen 1.5-MoE's late-stage layers (20–24) generate a gain of 1.45 per layer, compared to Qwen 1.5-7B's 0.68 per layer in layers 25–32. The efficiency stems from task-specialized routing and cross-expert collaboration, contrasting with dense models' uniform parameter utilization. The exceptional late-stage performance of Mistral and its resonance with Mixtral, however, indicate that initialization strategies (Lo et al., 2025) and architectural synergies may further modulate gain dynamics beyond conventional dense/MoE dichotomies.

3.3 Cross-Model Comparison of Stage Contributions

Having established the stage-wise processing patterns, we next compare their contributions across architectures. To quantify architectural differences, Figure 3 and Table 1 compare cumulative contributions across early, mid-, and late stages.

Mid-stage dominance in MoE Qwen 1.5-MoE achieves 41.7% mid-stage contribution (absolute gain: 2.93), surpassing Qwen 1.5-7B's 37.5% (2.56). OLMoE's shallow design further amplifies mid-stage reliance (52.2% contribution), but limited depth restricts total gain (4.79 vs. Qwen 1.5-MoE's 7.05).

Late-stage amplification in deep MoE Qwen 1.5-MoE's 24-layer architecture enables extended refinement, with 50.7% late-stage gain (3.57) versus Qwen 1.5-7B's 47.6% (3.25). OLMoE's 16-layer design results in rapid late-stage decay (12.5% contribution post-layer 12), highlighting the trade-off between depth and efficiency.

Model Config	HIT@10	MRR
Only Top1	0	0
Only Top2	0	0
Only Shared	0.03	0.01
Shared + 2nd	0.04	0.01
Shared + Top1	0.82	0.59
Shared + Top2	0.83	0.63
Shared + Top4 (default)	0.85	0.63

Table 2: Experimental results for different Qwen 1.5-MoE configurations, highlighting the effects of the shared expert and top-1 routing.

Model Config	HIT@10	MRR
Top0	0.00	0.00
Top1	0.86	0.67
Top2 (default)	0.90	0.73

Table 3: Experimental results for different Mixtral configurations.

Architectural impact on processing strategies Deep MoE models (e.g., Qwen 1.5-MoE) prioritize collaborative refinement in late layers, while shallow MoE architectures (e.g., OLMoE) focus on rapid mid-stage decisions. Dense models' dispersed gains hinder task specialization—Qwen 1.5-7B requires 32 layers to achieve comparable total gains, indicating inefficiency in cross-layer integration.

These results underscore that MoE's efficiency arises from stage-specialized processing, yet its effectiveness depends on balancing depth and expert collaboration. A detailed analysis of robustness implications (e.g., expert blocking effects) is provided in § 5.2.

4 Expert Collaboration: Basic-Refinement via Semantic Routing

This section investigates the collaborative mechanisms between shared and routed experts in MoE models. Through systematic ablation experiments, we validate the necessity of expert collaboration for robust knowledge processing (§ 4.1) and reveal how semantic-driven routing coordinates attention heads and experts into a hierarchical framework (§ 4.2).

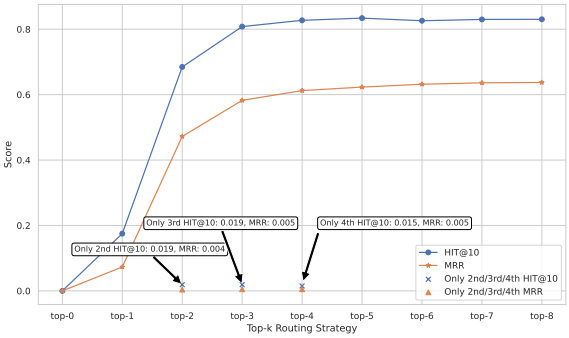

Figure 4: OLMoE's performance on knowledge prediction tasks when using different routing strategies.

4.1 Expert Role Specialization

Our ablation studies demonstrate that effective collaboration between shared and routed experts is fundamental to MoE performance. In both Qwen 1.5-MoE (Table 2)—which combines a shared expert with routed specialists—and OLMoE (Figure 4) and Mixtral (Table 3)—which relies solely on routed experts—isolating individual experts severely degrades model capability. For example, when only a single expert is activated, OLMoE's HIT@10 drops to a range of 0.15–0.19 with near-zero MRR, while Qwen 1.5-MoE achieves a mere HIT@10 of 0.03 when restricted to its shared experts and completely fails when limited to the top routed expert. These results underscore that complex knowledge tasks cannot be managed by isolated experts; instead, they require synergistic interactions between generalists and specialists.

A critical observation is the diminishing returns of increasing expert participation. In OLMoE, activating three experts elevates HIT@10 from 0.18 (single expert) to 0.81, but adding a fourth expert yields only a marginal gain (0.81 to 0.83). Similarly, Qwen 1.5-MoE sees limited improvement when expanding beyond its shared experts and top routed expert—HIT@10 rises from 0.82 to 0.85 and MRR from 0.59 to 0.63. This suggests the existence of an "effective expert threshold": core knowledge is predominantly captured by a small subset of experts, with additional modules contributing minimally.

Architectural differences further amplify these patterns. Qwen 1.5-MoE's shared experts handles cross-domain fundamentals—such as entity recognition and syntactic parsing—freeing routed experts to specialize in domain-specific refinement. This division allows Qwen 1.5-MoE to achieve high performance with minimal experts while maintaining fault tolerance; blocking a routed expert reduces HIT@10 by only 3% (0.85 to 0.82). In contrast, OLMoE's lack of a shared expert forces it to activate multiple routed experts (typically three) for comparable accuracy, making it vulnerable to routing failures. For instance, blocking a critical expert in OLMoE degrades MRR by over 30%, as shown in Figure 4.

These findings collectively support a "basic-refinement" framework for MoE design (as shown in Figure 1): (1) Shared experts act as generalists, providing robust foundational processing. (2) Routed experts specialize in refining domain-specific knowledge. This hierarchical collaboration not only enhances efficiency but also improves interpretability, offering practical guidance for optimizing MoE architectures under computational constraints. For instance, prioritizing shared experts in early layers and routed experts in deeper layers could balance performance and resource usage—a strategy validated by Qwen 1.5-MoE's success.

4.2 Semantic-Driven Attention Head and Expert Selection Mechanism

In large-scale MoE models, the interaction between attention heads and experts operates as a tightly coordinated, semantic-driven collaboration. This mechanism ensures that task-specific knowledge is dynamically routed and refined across layers (Figure 1). Take Qwen 1.5-MoE as an example, the process begins in the early to mid-layers (layers 18–20), where attention heads such as $\mathcal{H}_{10}^{18}$ focus on encoding foundational semantic features—for example, identifying entities like "Canada" as a country. These features are stored and processed by activated experts (e.g., $\mathcal{E}_{38}^{18}$), which serve as repositories of domain-specific knowledge.

As processing progresses to layer 20, the routing mechanism enters a critical phase. Here, attention heads like $\mathcal{H}_{12}^{19}$ align token-level semantics with expert capabilities, selectively activating the top-4 specialists through gating probabilities (Eq. 2). For instance, the head $\mathcal{H}_{12}^{19}$ may prioritize expert $\mathcal{E}_{59}^{20}$) for tasks involving geographic relations, leveraging its stored knowledge of country attributes. This semantic alignment minimizes redundant computations while maximizing task relevance.

In the deeper layers (21–23), the collaboration shifts toward refinement. Specialized experts such as $\mathcal{E}_{48}^{21}$) work under the guidance of attention heads like $\mathcal{H}_{13}^{22}$) to generate precise outputs—for exam-

ple, linking "Canada" to its capital "Ottawa". To validate this hierarchical pipeline, we analyzed temporal correlations between attention heads and experts. The results reveal statistically significant synchronization: the correlation coefficient between $\mathcal{E}_{10}^{18}$ and $\mathcal{E}_{59}^{20}$ reaches $r = 0.68 \quad (p < 0.001)$, while $\mathcal{H}_{15}^{23}$ and $\mathcal{E}_{20}^{23}$ exhibit $r = 0.62 \quad (p < 0.01)$ (Table 4). These findings confirm that attention heads actively steer expert selection in a semantics-aware manner, contrasting sharply with dense models' static parameter usage.

4.3 Causality Analysis

While the above temporal correlations provide strong evidence of semantic coordination between attention heads and experts, we recognize that correlation alone does not establish a causal relationship. To verify that attention heads actively drive expert selection, we designed and ran three targeted interventions:

Attention Head Suppression We selectively suppressed the activity of head $\mathcal{H}_{12}^{19}$ in Qwen during inference. This intervention reduced the average gating probability for the top-4 experts by 54% and led to a 29% drop in mean reciprocal rank (MRR), demonstrating that disabling a single key head disrupts downstream expert routing.

Expert Forcing In a complementary experiment, we forcibly activated expert $\mathcal{E}_{59}^{20}$ regardless of the gating probabilities. This intervention recovered 85% of the original MRR, indicating that reinstating the correct expert can largely restore model performance even when upstream signals are perturbed.

Path Analysis via Integrated Gradients We applied Integrated Gradients to trace the decision path from attention heads to experts. On average, 28% of the attribution for expert activations (e.g. the route $\mathcal{H}_{13}^{22} \rightarrow \mathcal{E}_{48}^{21}$) could be directly linked back to specific attention heads, quantifying their direct influence in the routing computation.

Together, these causal interventions confirm that attention heads do not merely co-activate with experts but in fact guide their selection in a semantics-aware, hierarchical pipeline.

5 MoE Knowledge Localization: Trade-offs and Robustness

In this section, we investigate the phenomenon of neuron-level knowledge localization in MoE models. For each knowledge type, identify the 100 most important neurons, determine which experts they belong to, and then we list the top 5 experts (i.e. the five experts containing the greatest number of those 100 neurons). Here, our analysis is more concentrated on fine-grained experts, whose specialization trends are particularly pronounced (Wang et al., 2024); for coarse-grained MoE models such as Mixtral, we provide a detailed description in Appendix B. We provide insights into expert specialization (§ 5.1) and investigate how architectural design affects the robustness and redundancy of the model, especially under expert failure scenarios (§ 5.2). Through systematic expert-blocking experiments, we reveal critical trade-offs between specialization depth, task sensitivity, and fault tolerance.

Relation	High-Imp. Attention Head	High-Freq. Routed Expert
capital	$\mathcal{H}_{13}^{22}, \mathcal{H}_{10}^{18}, \mathcal{H}_{15}^{23}$	$\mathcal{E}_{59}^{20}, \mathcal{E}_{48}^{21}, \mathcal{E}_{14}^{21}$
language	$\mathcal{H}_{13}^{22}, \mathcal{H}_{10}^{18}, \mathcal{H}_{15}^{23}$	$\mathcal{E}_{59}^{20}, \mathcal{E}_{20}^{23}, \mathcal{E}_{20}^{21}$
birthplace	$\mathcal{H}_{13}^{22}, \mathcal{H}_{10}^{18}, \mathcal{H}_{12}^{19}$	$\mathcal{E}_{59}^{20}, \mathcal{E}_{48}^{21}, \mathcal{E}_{38}^{18}$

Table 4: Top 3 attention heads and experts in Qwen 1.5-MoE.

5.1 Expert Specialization and Architectural Trade-offs

As shown in Table 5, our findings reveal that certain experts consistently rank among the top five across various knowledge categories such as name_birthplace, country_capital_city, and country_language.

The specialization of experts in MoE models is profoundly influenced by architectural depth. In deeper architectures like Qwen 1.5-MoE, experts adopt a clear hierarchical division of labor. Take "*The capital of Canada is the city of*" as an example. Early layers host generalists such as $\mathcal{E}_{59}^{20}$, which extract broad semantic features—for example, recognizing "Canada" as a geographic entity. Subsequent layers deploy specialists like $\mathcal{E}_{20}^{21}$, which refine these features into precise attributes (e.g., mapping "Canada" to "Ottawa"). This stratification is evident in tasks such as geographic reasoning: when processing the input "The capital of Canada is the city of", $\mathcal{E}_{59}^{20}$ first isolates the country context, while $\mathcal{E}_{20}^{21}$ generates the answer "Ottawa" through attribute association.

In contrast, shallow architectures like OLMoE

Relation	Model	Top5 Routed Experts	MRR (Excl. Top$n$ Experts)			
			-	Top1	Top5	Top10
name_birthplace	Qwen 1.5-MoE	$\mathcal{E}_{59}^{20}, \mathcal{E}_{20}^{23}, \mathcal{E}_{42}^{19}, \mathcal{E}_{37}^{22}, \mathcal{E}_{13}^{16}$	0.85	0.84	0.83	0.81
	OLMoE	$\mathcal{E}_{56}^{13}, \mathcal{E}_{49}^{15}, \mathcal{E}_{2}^{14}, \mathcal{E}_{40}^{12}, \mathcal{E}_{47}^{12}$	0.82	0.80	0.80	0.60
country_capital_city	Qwen 1.5-MoE	$\mathcal{E}_{59}^{20}, \mathcal{E}_{48}^{21}, \mathcal{E}_{1}^{22}, \mathcal{E}_{14}^{21}, \mathcal{E}_{38}^{18}$	0.71	0.68	0.68	0.40
	OLMoE	$\mathcal{E}_{56}^{13}, \mathcal{E}_{40}^{12}, \mathcal{E}_{2}^{14}, \mathcal{E}_{52}^{13}, \mathcal{E}_{12}^{14}$	1.0	1.0	0.76	0.76
country_language	Qwen 1.5-MoE	$\mathcal{E}_{16}^{22}, \mathcal{E}_{59}^{20}, \mathcal{E}_{20}^{23}, \mathcal{E}_{20}^{21}, E_{3}^{19}$	0.94	0.92	0.92	0.92
	OLMoE	$\mathcal{E}_{7}^{13}, \mathcal{E}_{3}^{14}, \mathcal{E}_{49}^{15}, \mathcal{E}_{0}^{12}, \mathcal{E}_{56}^{13}$	0.96	0.92	0.87	0.68
fruit_inside_color	Qwen 1.5-MoE	$\mathcal{E}_{15}^{19}, \mathcal{E}_{38}^{22}, \mathcal{E}_{1}^{20}, \mathcal{E}_{15}^{22}, \mathcal{E}_{10}^{18}$	0.74	0.68	0.63	0.60
	OLMoE	$\mathcal{E}_{11}^{12}, \mathcal{E}_{8}^{14}, \mathcal{E}_{45}^{15}, \mathcal{E}_{32}^{12}, \mathcal{E}_{59}^{13}$	0.76	0.60	0.52	0.41
object_superclass	Qwen 1.5-MoE	$\mathcal{E}_{11}^{22}, \mathcal{E}_{25}^{21}, \mathcal{E}_{25}^{21}, \mathcal{E}_{1}^{20}, \mathcal{E}_{15}^{19}$	0.83	0.82	0.80	0.79
	OLMoE	$\mathcal{E}_{8}^{14}, \mathcal{E}_{59}^{13}, \mathcal{E}_{38}^{15}, \mathcal{E}_{53}^{14}, \mathcal{E}_{59}^{14}$	0.80	0.75	0.70	0.66

Table 5: Top 5 experts for each knowledge type, along with the model's MRR after excluding the top 1, top 5, and top 10 of those experts. The complete experimental results can be found in Appendix B.

face inherent constraints. With only 16 layers, experts such as $\mathcal{E}_{13}^{56}$must balance multiple roles, handling both geographic tasks (e.g., country-language mappings) and object attributes (e.g., fruit_inside_color). While this promotes cross-task generalization, it sacrifices fine-grained precision. For instance, $\mathcal{E}_{8}^{14}$ in OLMoE processes both fruit colors and object hierarchies, leading to blurred functional boundaries compared to Qwen 1.5-MoE's specialized experts.

These architectural differences further impact robustness. Deep models leverage shared experts and layered redundancy to mitigate failures: blocking $\mathcal{E}_{20}^{21}$ in Qwen 1.5-MoE causes only a marginal MRR drop ($0.85 \rightarrow 0.84$) for geographic tasks. Shallow models, however, lack such safeguards. Blocking the critical expert $\mathcal{E}_{13}^{56}$ in OLMoE degrades MRR by 30%, underscoring the vulnerability of architectures that prioritize breadth over depth. This trade-off highlights a fundamental design principle: deeper MoE models achieve robustness through hierarchical specialization, while shallower models rely on expert versatility at the cost of precision.

5.2 Robustness and Redundancy in MoE Architectures

Table 5 shows the changes in MRR of Qwen 1.5-MoE and OLMoE models after blocking top-n experts under different knowledge types. Experimental results show that as the Top-1, Top-5, and Top-10 experts are blocked, Qwen 1.5-MoE and OLMoE show different robustness in different tasks.

The interplay between expert specialization and architectural depth profoundly impacts model robustness. For Qwen 1.5-MoE, its 24-layer design and shared expert integration enable distributed knowledge storage, buffering against expert failures. Blocking the top 10 experts in general tasks like name_birthplace causes only a marginal MRR drop ($0.85 \rightarrow 0.81$), as redundant general experts (e.g., $\mathcal{E}_{59}^{20}$) compensate for losses. However, geographic tasks like country_capital_city exhibit fragility: blocking core experts (e.g., $\mathcal{E}_{48}^{21}$) slashes MRR by 43% ($0.71 \rightarrow 0.40$), highlighting their reliance on specialized modules.

In contrast, OLMoE's shallow 16-layer architecture lacks such safeguards. While its routing mechanism dynamically selects alternative experts to maintain performance when blocking top 1–5 experts (MRR remains 1.0 for country_capital_city), deeper dependencies overwhelm its redundancy capacity. Blocking the top 10 experts degrades MRR to 0.76, with geographic tasks like country_language suffering a 30% decline ($0.96 \rightarrow 0.68$). This stark contrast underscores that depth enables hierarchical redundancy, whereas shallow models trade transient adaptability for long-term fragility.

Furthermore, task sensitivity further dictates robustness requirements. **Core-sensitive tasks** (e.g., geography) demand concentrated expertise. Qwen 1.5-MoE's late-layer specialists (e.g., $\mathcal{E}_{48}^{21}$) are irreplaceable, necessitating redundancy through shared experts. **Distributed-tolerant tasks** (e.g., object attributes) leverage broader expert participation. Qwen 1.5-MoE's MRR drops mildly ($0.83 \rightarrow 0.79$) when blocking experts for object_superclass, while OLMoE plummets ($0.80 \rightarrow 0.66$) due to overburdened generalists.

These findings inform actionable design princi-

ples: (1) Deep MoE architectures should prioritize shared experts in early layers for cross-task redundancy and allocate specialists to late layers for core-sensitive tasks. (2) Shallow MoEs must balance expert versatility and routing adaptability, though their limited depth inherently restricts robustness. (3) By aligning architectural depth with task requirements, MoEs can optimize both performance and reliability—a strategy exemplified by Qwen 1.5-MoE's resilience and OLMoE's adaptability trade-offs.

6 Related Work

Mixture of Experts The Mixture of Experts (MoE) framework, introduced by Jacobs et al. (1991), dynamically activates specialized subnetworks for inputs. Early applications in language modeling used LSTMs (Shazeer et al., 2017), but recent work integrates MoE into Transformer FFN layers (Vaswani, 2017), scaling parameters via sparse routing (e.g., top-k gating (Fedus et al., 2021; Lepikhin et al., 2021; Roller et al., 2021; Dai et al., 2022b; Zeng et al., 2024; Xue et al., 2024)). Innovations like DeepSeekMoE (Dai et al., 2024) and Qwen-MoE (Team, 2024) enhance specialization through shared experts, balancing efficiency and performance.

Knowledge Attribution Studies on how models store knowledge reveal that Transformer MLPs act as key-value memories (Geva et al., 2021; Dai et al., 2022a; Meng et al., 2022; Wu et al., 2023; Xu et al., 2023; Meng et al., 2023), with knowledge accumulating across layers (Geva et al., 2022; Lv et al., 2024). While methods like knowledge circuits (Yao et al., 2024) and neuron-level attribution (Yu and Ananiadou, 2024) interpret dense models, sparse MoE architectures—with dynamic routing and expert collaboration—remain underexplored. Our work bridges this gap, offering the first comprehensive analysis of MoE interpretability.

7 Conclusion

In this paper, we propose a cross-level knowledge attribution algorithm for MoEs, bridging the interpretability gap between sparse and dense architectures. Through comparative analysis, we uncover MoE's unique efficiency patterns—such as "mid-stage activation, late-stage amplification" and a "basic-refinement" collaboration framework—where shared experts handle general tasks while routed experts specialize in refinement. We further validate a semantic-driven routing mechanism: attention heads coordinate expert selection via high temporal correlations, enabling task-aware processing.

Our findings offer actionable guidelines for MoE design:

- Prioritize shared experts in early layers to provide cross-task redundancy and foundational processing.
- Allocate routed experts to deeper layers for domain-specific refinement, balancing specialization and efficiency.
- Optimize routing depth based on task sensitivity: Deploy deeper architectures for core-sensitive tasks (e.g., geography) and shallower models for distributed-tolerant tasks (e.g., object attributes).
- Enhance semantic alignment between attention heads and experts to improve routing robustness.

These principles address critical challenges in MoE scalability and reliability, offering a pathway to deploy efficient, interpretable models in resource-constrained scenarios.

Acknowledgments

This work was supported by the CAAI-Ant Group Research Fund; Guangdong Provincial Department of Education Project (Grant No.2024KQNCX028); Scientific Research Projects for the Higher-educational Institutions (Grant No.2024312096), Education Bureau of Guangzhou Municipality; Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3957), Education Bureau of Guangzhou Municipality.

Limitations

Our analysis is confined to 7B-parameter, static-routing MoE models, so it remains to be seen whether much larger (e.g., 100 B+) (DeepSeek-AI et al., 2024; Sun et al., 2024) or fully dynamic MoE (Huang et al., 2024; Guo et al., 2025) architectures will exhibit the same specialization patterns. We also omit retrieval-augmented MoEs (e.g., Monet (Park et al., 2025)), whose external-knowledge coupling adds confounding factors beyond our scope.

References

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Damai Dai, Chengqi Deng, Chenggang Zhao, R.x. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y.k. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297, Bangkok, Thailand. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022a. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022b. StableMoE: Stable routing strategy for mixture of experts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7085–7095, Dublin, Ireland. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint*.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer

Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. 2025. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. *Preprint*, arXiv:2405.14297.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. 2024. Harder task needs more experts: Dynamic routing in MoE models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12883–12895, Bangkok, Thailand. Association for Computational Linguistics.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

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

Ang Lv, Yuhan Chen, Kaiyi Zhang, Yulong Wang, Lifeng Liu, Ji-Rong Wen, Jian Xie, and Rui Yan. 2024. Interpreting key mechanisms of factual recall in transformer-based language models. *arXiv preprint arXiv:2403.19521*.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 338–345.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. *Preprint*, arXiv:2210.07229.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. 2024. Olmoe: Open mixture-of-experts language models. *Preprint*, arXiv:2409.02060.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.

Jungwoo Park, Ahn Young Jin, Kee-Eung Kim, and Jaewoo Kang. 2025. Monet: Mixture of monosemantic experts for transformers. In *The Thirteenth International Conference on Learning Representations*.

Stephen Roller, Sainbayar Sukhbaatar, arthur szlam, and Jason Weston. 2021. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems*, volume 34, pages 17555–17566. Curran Associates, Inc.

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*.

Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, et al. 2023. Mixture-of-experts meets instruction tuning: A winning combination for large language models. *arXiv preprint arXiv:2305.14705*.

Xingwu Sun, Yanfeng Chen, Yiqing Huang, Ruobing Xie, Jiaqi Zhu, Kai Zhang, Shuaipeng Li, Zhen Yang, Jonny Han, Xiaobo Shu, Jiahao Bu, Zhongzhi Chen, Xuemeng Huang, Fengzong Lian, Saiyong Yang, Jianfeng Yan, Yuyuan Zeng, Xiaoqin Ren, Chao Yu, Lulu Wu, Yue Mao, Tao Yang, Suncong Zheng, Kan Wu, Dian Jiao, Jinbao Xue, Xipeng Zhang, Decheng Wu, Kai Liu, Dengpeng Wu, Guanghui Xu, Shaohua Chen, Shuang Chen, Xiao Feng, Yigeng Hong, Junqiang Zheng, Chengcheng Xu, Zongwei Li, Xiong Kuang, Jianglu Hu, Yiqi Chen, Yuchi Deng, Guiyang Li, Ao Liu, Chenchen Zhang, Shihui Hu, Zilong Zhao, Zifan Wu, Yao Ding, Weichao Wang, Han Liu, Roberts Wang, Hao Fei, Peijie She, Ze Zhao, Xun Cao, Hai Wang, Fusheng Xiang, Mengyuan Huang, Zhiyuan Xiong, Bin Hu, Xuebin Hou, Lei Jiang, Jiajia Wu, Yaping Deng, Yi Shen, Qian Wang, Weijie Liu, Jie Liu, Meng Chen, Liang Dong, Weiwen Jia, Hu Chen, Feifei Liu, Rui Yuan, Huilin Xu, Zhenxiang Yan, Tengfei Cao, Zhichao Hu, Xinhua Feng, Dong Du, Tinghao She, Yangyu Tao, Feng Zhang, Jianchen Zhu, Chengzhong Xu, Xirui Li, Chong Zha, Wen Ouyang, Yinben Xia, Xiang Li, Zekun He, Rongpeng Chen, Jiawei Song, Ruibin Chen, Fan Jiang, Chongqing Zhao, Bo Wang, Hao Gong, Rong Gan, Winston Hu, Zhanhui Kang, Yong Yang, Yuhong Liu, Di Wang, and Jie Jiang. 2024. Hunyuan-large: An open-source moe model with 52 billion activated parameters by tencent. *Preprint*, arXiv:2411.02265.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Yu Wu. 2024. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 784–801, Miami, Florida, USA. Association for Computational Linguistics.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. DEPN: Detecting and editing privacy neurons in pretrained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2875–2886, Singapore. Association for Computational Linguistics.

Shaoyang Xu, Junzhuo Li, and Deyi Xiong. 2023. Language representation projection: Can we transfer factual knowledge across languages in multilingual language models? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3692–3702, Singapore. Association for Computational Linguistics.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. 2024. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024. Knowledge circuits in pretrained transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zeping Yu and Sophia Ananiadou. 2024. Neuron-level knowledge attribution in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3267–3280, Miami, Florida, USA. Association for Computational Linguistics.

Zhiyuan Zeng, Qipeng Guo, Zhaoye Fei, Zhangyue Yin, Yunhua Zhou, Linyang Li, Tianxiang Sun, Hang Yan, Dahua Lin, and Xipeng Qiu. 2024. Turn waste into worth: Rectifying top-k router of MoE. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13305–13320, Miami, Florida, USA. Association for Computational Linguistics.

A Experiment Settings

Models We evaluate multiple models to ensure robust conclusions. These include three dense models (Llama-7B (Touvron et al., 2023a), Qwen 1.5-7B (Bai et al., 2023)) and Mistral-7B (Jiang et al., 2023) and three Mixture-of-Experts (MoE) models (Qwen 1.5-MoE (Team, 2024), OLMoE (Muennighoff et al., 2024) and Mixtral 8x7B (Jiang et al., 2024)). Qwen 1.5-MoE is a 24-layer model where each MoE layer comprises a shared expert and 60 routing experts, employing a Top-4 routing strategy during inference. Mixtral has 32 layers, 8 experts per layer, no shared experts, employing a Top-2 routing strategy. In contrast, OLMoE is a shallower model with 16 layers, where each MoE layer contains 64 routing experts and employs a Top-8 routing mechanism. These structural differences among the MoE models provide complementary perspectives, enriching our analysis. Table 6 lists the specific configuration comparison of each model.

Datasets We use the dataset provided by LRE (Hernandez et al., 2024), which contains a diverse set of knowledge types, including linguistic patterns, common sense, factual information, and biases. A total of 12 distinct knowledge categories are selected for our experiments from the dataset (i.e., *adjective_antonym, word_first_letter, word_last_letter relations, object_superclass, fruit_inside_color, work_location relations, country_language, country_capital, name_birthplace, name_religion, occupation_age,* and *occupation_gender*). This dataset serves two purposes: it provides explicit knowledge for the model and is also used to evaluate its zero-shot performance across varied knowledge domains.

Metrics We assess model performance using two standard metrics: HIT@10 and Mean Reciprocal Rank (MRR). HIT@10 measures the proportion of test instances for which the correct answer is ranked among the top 10 predictions:

$$\text{HIT@10} = \frac{1}{|V|}\sum_{i=1}^{|V|}\mathbb{I}(\text{rank}_i \leq 10), \qquad (10)$$

where $\mathbb{I}(\cdot)$ is the indicator function, returning 1 if the condition is satisfied and 0 otherwise, and rank_i denotes the rank position of the correct answer for the i-th instance.

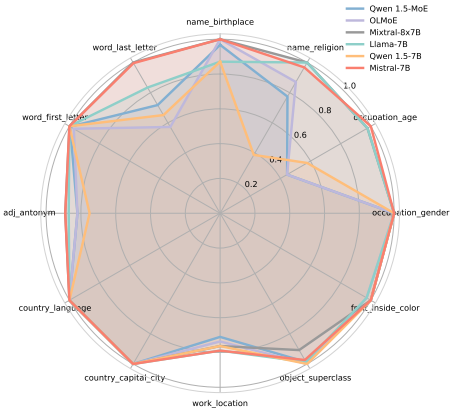

Figure 5: HIT@10 scores of four models on 12 different relations.

MRR computes the average of the reciprocal ranks of the correct answers:

$$\text{MRR} = \frac{1}{|V|}\sum_{i=1}^{|V|}\frac{1}{\text{rank}_i}. \qquad (11)$$

While HIT@10 emphasizes the model's ability to retrieve correct answers within the top predictions, MRR provides a more sensitive measure by giving higher scores to cases where the correct answer is ranked highly. Together, these metrics offer complementary insights into the overall ranking performance of the model.

Results Figure 5 shows the HIT@10 of different models on 12 tasks. It can be seen that the performance of different models varies greatly in some types. Therefore, when analyzing the experimental results, we mainly focus on the relations with similar performance to avoid the influence of the model's internal parameter knowledge on the effect of knowledge attribution.

B Detailed Analysis of Coarse/Fine-Grained Routed Experts

We report the full expert-isolation results in Table 8. Figure 7 visualizes the heat map of the MRR drop of the models under different settings.As expected, the two fine-grained MoE models suffer precipitous drops in performance when their top specialists are blocked—reaffirming the strong localization of functionality we described in § 5.2. By contrast, Mixtral—a coarse-grained MoE with only eight experts per layer and a top-2 routing policy—maintains near-identical accuracy even when its ten most "important" experts are silenced.

Model Name	Layers	FFN / Expert Dimension	Number of Experts	Routing Strategy	Shared Expert	Shared Expert Dimension
LLama-7B	32	11008	-	-	-	-
Qwen 1.5-7B	32	11008	-	-	-	-
Mistral-7B	32	14336	-	-	-	-
Qwen1.5-MoE-A2.7B	24	1408	60	Top-4	Yes	5632
OLMoE-1B-7B	16	1024	64	Top-8	-	-
Mixtral-8x7B	32	14336	8	Top-2	-	-

Table 6: Model Configuration.

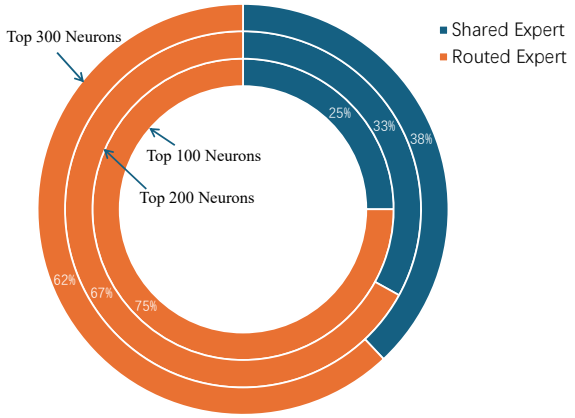

Figure 6: Type distribution of neurons in the top 100, 200, and 300 of Qwen 1.5-MoE importance scores.

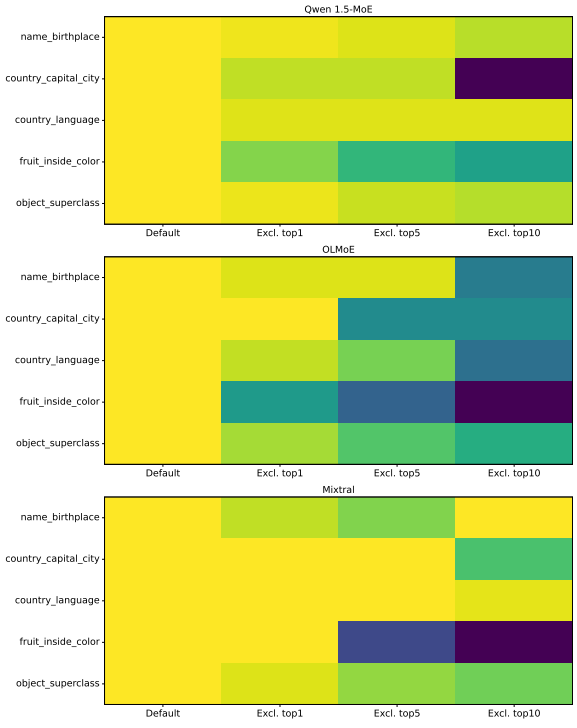

Figure 7: Robustness comparison heat map. The darker the color, the greater the drop in MRR.

This resilience stems from Mixtral's relatively dense routing: with two experts activated per token, each expert has a $1/4$ chance of selection, compared to just $1/15$ in Qwen 1.5-MoE and $1/8$ in OLMoE. Over long sequences, this routing density means that almost every expert participates, distributing representational capacity broadly rather than concentrating it in a few specialists. Since the Mixtral authors do not explicitly detail their expert initialization, we tentatively infer—based on related research (Lo et al., 2025)—that its experts may be seeded by copying FFN weights from the dense Mistral model, a practice that would likely homogenize their behaviors and impede the development of fine-grained specialization.

Taken together, these findings highlight a fundamental trade-off in MoE design. Coarse-grained architectures like Mixtral effectively amplify model capacity and fault tolerance by maximizing activation coverage, but they do so at the expense of individual expert specialization. Fine-grained MoEs, on the other hand, cultivate highly specialized experts—yielding sharp performance gains on focused tasks—but are more vulnerable to expert failure. Future work might explore hybrid schemes that combine a base level of broad activation with targeted specialization, for instance by increasing expert count, adjusting routing sparsity, or employing specialized initialization and regularization to encourage functional diversity among experts.

C Cross-Knowledge Neuron Dynamics

In the context of shared experts, we observe a significant overlap of neurons for different samples within the same relation (Table 7). For instance, in the "capital" task, neurons responsible for predicting the capitals of various countries often share common activations. This suggests that neurons

Category	Top-5 Routed Shared Expert Neurons
birthplace	$\mathcal{N}^{20}_{1044}, \mathcal{N}^{16}_{204}, \mathcal{N}^{21}_{711}, \mathcal{N}^{17}_{123}, \mathcal{N}^{19}_{641}$
capital	$\mathcal{N}^{20}_{5}, \mathcal{N}^{15}_{1270}, \mathcal{N}^{15}_{415}, \mathcal{N}^{9}_{1060}, \mathcal{N}^{18}_{637}$
language	$\mathcal{N}^{18}_{1137}, \mathcal{N}^{18}_{232}, \mathcal{N}^{17}_{682}, \mathcal{N}^{15}_{647}, \mathcal{N}^{19}_{322}$
fruit	$\mathcal{N}^{20}_{947}, \mathcal{N}^{16}_{413}, \mathcal{N}^{15}_{173}, \mathcal{N}^{19}_{1158}, \mathcal{N}^{19}_{900}$
object	$\mathcal{N}^{17}_{202}, \mathcal{N}^{18}_{1145}, \mathcal{N}^{20}_{631}, \mathcal{N}^{21}_{1262}, \mathcal{N}^{20}_{359}$

Table 7: Top-5 shared expert neurons for each relation. For example, $\mathcal{N}^{20}_{1044}$ represents the 1044th neuron of the shared expert in the 20th layer.

can store knowledge specific to a given relation and share it across different instances. However, the overlap of neurons across different relations remains minimal. For example, the shared neuron overlap between the "capital" and "language" tasks is typically less than 5%. This indicates that the role of shared experts is task-specific, with distinct mechanisms for knowledge storage and processing across tasks.

Further analysis, as shown in Table 8, reveals that the effect significantly increases after blocking the top 10 experts for certain relations (e.g., the MRR for the occupation_age relation in Qwen 1.5-MoE changes from 0.22 to 0.36). This counterintuitive phenomenon contrasts with the behavior observed in other relations, where blocking commonly used experts leads to a decrease in performance or fluctuations around baseline values. However, it is important to note that the model's baseline MRR on this particular task is quite low and close to random, meaning that the observed increase does not indicate an issue with our method.

Moreover, when analyzing the top 100 neurons, we find that approximately 25% of these neurons belong to routed experts, while the remaining 75% are associated with shared experts, as shown in Figure 6. However, as we expand the analysis to the top 300 neurons, the proportion of routed experts increases to 38%. This suggests that more expert collaboration is required for handling complex tasks, with the routing mechanism dynamically expanding its activation range. As task complexity increases, the routing mechanism activates additional experts, creating a dynamic collaboration chain that connects core experts with auxiliary experts. This mechanism enables the model to flexibly coordinate knowledge from a broader set of experts, ultimately improving its performance on more complex tasks.

Relation	Model	Top5 Routed Experts	MRR (Excl. Top$n$ Experts)			
			-	Top1	Top5	Top10
country_capital_city	Qwen 1.5-MoE	$\mathcal{E}_{59}^{20}, \mathcal{E}_{48}^{21}, \mathcal{E}_{1}^{22}, \mathcal{E}_{14}^{21}, \mathcal{E}_{38}^{18}$	0.71	0.68	0.68	0.40
	OLMoE	$\mathcal{E}_{56}^{13}, \mathcal{E}_{40}^{12}, \mathcal{E}_{2}^{14}, \mathcal{E}_{52}^{13}, \mathcal{E}_{12}^{14}$	1.0	1.0	0.76	0.76
	Mixtral	$\mathcal{E}_{7}^{24}, \mathcal{E}_{0}^{25}, \mathcal{E}_{1}^{22}, \mathcal{E}_{0}^{26}, \mathcal{E}_{2}^{19}$	1.0	1.0	1.0	0.93
country_language	Qwen 1.5-MoE	$\mathcal{E}_{16}^{22}, \mathcal{E}_{59}^{20}, \mathcal{E}_{20}^{23}, \mathcal{E}_{20}^{21}, E_{3}^{19}$	0.94	0.92	0.92	0.92
	OLMoE	$\mathcal{E}_{7}^{13}, \mathcal{E}_{3}^{14}, \mathcal{E}_{49}^{15}, \mathcal{E}_{0}^{12}, \mathcal{E}_{56}^{13}$	0.96	0.92	0.87	0.68
	Mixtral	$\mathcal{E}_{5}^{22}, \mathcal{E}_{4}^{24}, \mathcal{E}_{6}^{19}, \mathcal{E}_{7}^{25}, \mathcal{E}_{0}^{21}$	0.98	0.98	1.0	0.97
adj_antonym	Qwen 1.5-MoE	$\mathcal{E}_{27}^{20}, \mathcal{E}_{15}^{22}, \mathcal{E}_{59}^{22}, \mathcal{E}_{49}^{16}, \mathcal{E}_{4}^{21}$	0.60	0.57	0.50	0.53
	OLMoE	$\mathcal{E}_{24}^{11}, \mathcal{E}_{49}^{13}, \mathcal{E}_{32}^{12}, \mathcal{E}_{36}^{12}, \mathcal{E}_{25}^{14}$	0.67	0.64	0.65	0.58
	Mixtral	$\mathcal{E}_{0}^{20}, \mathcal{E}_{6}^{19}, \mathcal{E}_{5}^{23}, \mathcal{E}_{6}^{31}, \mathcal{E}_{5}^{21}$	0.75	0.75	0.74	0.73
word_first_letter	Qwen 1.5-MoE	$\mathcal{E}_{7}^{19}, \mathcal{E}_{41}^{16}, \mathcal{E}_{49}^{16}, \mathcal{E}_{41}^{18}, \mathcal{E}_{45}^{19}$	0.96	0.95	0.93	0.92
	OLMoE	$\mathcal{E}_{39}^{14}, \mathcal{E}_{17}^{15}, \mathcal{E}_{45}^{12}, \mathcal{E}_{32}^{12}, \mathcal{E}_{60}^{15}$	0.79	0.75	0.68	0.48
	Mixtral	$\mathcal{E}_{7}^{30}, \mathcal{E}_{7}^{29}, \mathcal{E}_{0}^{28}, \mathcal{E}_{4}^{27}, \mathcal{E}_{3}^{28}$	1.0	1.0	1.0	0.99
word_last_letter	Qwen 1.5-MoE	$\mathcal{E}_{7}^{19}, \mathcal{E}_{41}^{16}, \mathcal{E}_{49}^{16}, \mathcal{E}_{45}^{19}, \mathcal{E}_{41}^{18}$	0.27	0.25	0.12	0.12
	OLMoE	$\mathcal{E}_{7}^{14}, \mathcal{E}_{32}^{12}, \mathcal{E}_{17}^{15}, \mathcal{E}_{7}^{11}, \mathcal{E}_{62}^{10}$	0.12	0.12	0.10	0.08
	Mixtral	$\mathcal{E}_{7}^{30}, \mathcal{E}_{7}^{29}, \mathcal{E}_{0}^{28}, \mathcal{E}_{4}^{27}, \mathcal{E}_{7}^{24}$	0.95	0.95	0.91	0.68
name_birthplace	Qwen 1.5-MoE	$\mathcal{E}_{59}^{20}, \mathcal{E}_{20}^{23}, \mathcal{E}_{42}^{19}, \mathcal{E}_{37}^{22}, \mathcal{E}_{13}^{16}$	0.85	0.84	0.83	0.81
	OLMoE	$\mathcal{E}_{56}^{13}, \mathcal{E}_{49}^{15}, \mathcal{E}_{2}^{14}, \mathcal{E}_{40}^{12}, \mathcal{E}_{47}^{12}$	0.82	0.80	0.80	0.60
	Mixtral	$\mathcal{E}_{1}^{18}, \mathcal{E}_{1}^{22}, \mathcal{E}_{3}^{25}, \mathcal{E}_{4}^{29}, \mathcal{E}_{2}^{31}$	0.87	0.85	0.83	0.87
name_religion	Qwen 1.5-MoE	$\mathcal{E}_{30}^{20}, \mathcal{E}_{20}^{23}, \mathcal{E}_{13}^{16}, \mathcal{E}_{53}^{20}, \mathcal{E}_{38}^{18}$	0.36	0.45	0.32	0.37
	OLMoE	$\mathcal{E}_{14}^{13}, \mathcal{E}_{37}^{12}, \mathcal{E}_{47}^{12}, \mathcal{E}_{1}^{13}, \mathcal{E}_{49}^{15}$	0.44	0.46	0.45	0.44
	Mixtral	$\mathcal{E}_{4}^{19}, \mathcal{E}_{5}^{20}, \mathcal{E}_{5}^{22}, \mathcal{E}_{7}^{24}, \mathcal{E}_{1}^{18}$	0.87	0.86	0.84	0.82
occupation_age	Qwen 1.5-MoE	$\mathcal{E}_{46}^{20}, \mathcal{E}_{48}^{19}, \mathcal{E}_{3}^{21}, \mathcal{E}_{10}^{18}, \mathcal{E}_{9}^{22}$	0.22	0.21	0.36	0.36
	OLMoE	$\mathcal{E}_{27}^{12}, \mathcal{E}_{26}^{13}, \mathcal{E}_{56}^{14}, \mathcal{E}_{62}^{15}, \mathcal{E}_{53}^{11}$	0.36	0.40	0.37	0.40
	Mixtral	$\mathcal{E}_{0}^{29}, \mathcal{E}_{4}^{30}, \mathcal{E}_{0}^{20}, \mathcal{E}_{6}^{31}, \mathcal{E}_{4}^{31}$	0.35	0.37	0.49	0.58
occupation_gender	Qwen 1.5-MoE	$\mathcal{E}_{42}^{22}, \mathcal{E}_{4}^{17}, \mathcal{E}_{29}^{19}, \mathcal{E}_{46}^{20}, \mathcal{E}_{10}^{18}$	0.95	0.96	0.94	0.96
	OLMoE	$\mathcal{E}_{27}^{12}, \mathcal{E}_{44}^{13}, \mathcal{E}_{1}^{13}, \mathcal{E}_{26}^{13}, \mathcal{E}_{14}^{15}$	0.95	0.86	0.84	0.55
	Mixtral	$\mathcal{E}_{0}^{23}, \mathcal{E}_{0}^{20}, \mathcal{E}_{6}^{19}, \mathcal{E}_{6}^{25}, \mathcal{E}_{0}^{29}$	0.97	1.0	0.97	0.97
fruit_inside_color	Qwen 1.5-MoE	$\mathcal{E}_{15}^{19}, \mathcal{E}_{38}^{22}, \mathcal{E}_{1}^{20}, \mathcal{E}_{15}^{22}, \mathcal{E}_{10}^{18}$	0.74	0.68	0.63	0.60
	OLMoE	$\mathcal{E}_{11}^{12}, \mathcal{E}_{8}^{14}, \mathcal{E}_{45}^{15}, \mathcal{E}_{32}^{12}, \mathcal{E}_{59}^{13}$	0.76	0.60	0.52	0.41
	Mixtral	$\mathcal{E}_{4}^{27}, \mathcal{E}_{2}^{26}, \mathcal{E}_{3}^{24}, \mathcal{E}_{5}^{23}, \mathcal{E}_{0}^{29}$	0.74	0.75	0.60	0.56
object_superclass	Qwen 1.5-MoE	$\mathcal{E}_{11}^{22}, \mathcal{E}_{25}^{21}, \mathcal{E}_{25}^{21}, \mathcal{E}_{1}^{20}, \mathcal{E}_{15}^{19}$	0.83	0.82	0.80	0.79
	OLMoE	$\mathcal{E}_{8}^{14}, \mathcal{E}_{59}^{13}, \mathcal{E}_{38}^{15}, \mathcal{E}_{53}^{14}, \mathcal{E}_{59}^{14}$	0.80	0.75	0.70	0.66
	Mixtral	$\mathcal{E}_{6}^{20}, \mathcal{E}_{3}^{31}, \mathcal{E}_{2}^{21}, \mathcal{E}_{6}^{19}, \mathcal{E}_{4}^{24}$	0.76	0.75	0.73	0.72
work_location	Qwen 1.5-MoE	$\mathcal{E}_{51}^{18}, \mathcal{E}_{8}^{22}, \mathcal{E}_{47}^{17}, \mathcal{E}_{27}^{19}, \mathcal{E}_{54}^{22}$	0.45	0.47	0.51	0.51
	OLMoE	$\mathcal{E}_{43}^{14}, \mathcal{E}_{0}^{12}, \mathcal{E}_{63}^{13}, \mathcal{E}_{13}^{13}, \mathcal{E}_{25}^{12}$	0.52	0.48	0.45	0.52
	Mixtral	$\mathcal{E}_{1}^{21}, \mathcal{E}_{5}^{20}, \mathcal{E}_{0}^{23}, \mathcal{E}_{0}^{24}, \mathcal{E}_{1}^{28}$	0.61	0.61	0.59	0.58

Table 8: Top-5 most frequently selected experts for each knowledge type, along with the MRR of the model after excluding the top 1, 5, and 10 experts.