Learned Routers Don't Learn: Statistical Evidence for Expert Miscalibration in Mixture-of-Experts Models

Young, 2026

March 24, 2026

Abstract

We present empirical evidence that learned routers in Mixture-of-Experts (MoE) transformer models are miscalibrated with respect to expert quality. Using a per-layer expert isolation methodology with log-probability scoring and rigorous multiple comparison correction (Benjamini-Hochberg step-up FDR), we demonstrate that: (1) experts have statistically significant domain specialization (207/896 expert-layer-domain combinations survive BH-FDR at $\alpha = 0.05$), (2) the learned router ignores this specialization (Fisher z-averaged Spearman $\rho = -0.017$ between natural routing probability and expert quality), and (3) a single expert (E2) is moderately preferred across all domains ($\sim$20% above uniform) despite never being the best expert for any domain tested. We propose semantic routing replacement and cross-model expert grafting as zero-training alternatives, and discuss implications for MoE architecture design. These findings suggest that learned routing does not correlate with expert quality in the marginal/isolation sense, opening new questions about whether current MoE training objectives adequately target individual expert specialization.

1 Introduction

Mixture-of-Experts (MoE) models have emerged as a promising direction for efficient scaling of transformer architectures. By sparse routing tokens through a subset of expert modules, MoE systems reduce computational cost while maintaining model capacity. However, MoE's effectiveness depends critically on one assumption: *learned routers learn to send tokens to the experts that produce the best outputs.*

The prevailing narrative treats expert routing as largely solved — a technical implementation detail optimized via auxiliary losses and load-balancing constraints. Yet this assumption has received surprisingly limited empirical scrutiny. Do routers actually learn expert quality? Or do they converge to some other objective that leaves expert specialization largely untapped?

We investigate this question using a per-layer expert isolation methodology. By forcing the router to attend to individual experts one at a time (at a single layer while maintaining normal routing elsewhere), we can measure each expert's marginal contribution to task performance, independent of the router's natural preferences. We then correlate these measured contributions with the router's actual activation probabilities.

Our key findings:

1. Experts *are* specialized: Domain-specific experts significantly improve performance on their respective domains (207/896 tests survive multiple comparison correction).

2. The router *does not exploit* this: Spearman correlation between routing probability and expert quality is near zero ($\rho = -0.017$ on average).

3. A single mediocre expert dominates routing across all domains, despite never being optimal.

4. This pattern holds across diverse domains (math, science, general knowledge, reasoning) and is not an artifact of our measurement methodology.

These findings do not necessarily mean MoE routing is useless. The router may optimize for expert *combinations* across layers rather than individual expert quality. However, they reveal a significant gap between the assumed behavior (routing based on expert quality) and observed behavior (routing indifferent to expert quality). We discuss implications and propose methods to exploit expert specialization without retraining.

2 Related Work

2.1 MoE Routing and Architecture

Switch Transformers (1) introduced simplified sparse routing to MoE and identified load balancing as critical for training stability. However, load balancing objectives may conflict with routing based on expert quality — a point supported by our finding that routers remain miscalibrated even after full training.

Expert Choice routing (2) proposes letting experts choose tokens rather than tokens choosing experts, arguing that expert-centric routing aligns better with expert specialization. Our near-zero Spearman correlation provides empirical support for this direction.

DeepSeekMoE (3) introduces fine-grained expert segmentation and shared/routed expert design. Our per-layer analysis reveals that expert quality varies dramatically by layer position, suggesting that per-layer routing flexibility (as in DeepSeekMoE) may help exploit discovered specialization.

2.2 Expert Analysis and Collapse

Prior work has studied expert collapse and load imbalance (4; 5). Our contribution complements this literature by measuring not just activation balance but alignment between activations and expert quality.

2.3 This Work's Contribution

To our knowledge, this is the first systematic study of Spearman correlation between learned routing probabilities and measured expert quality, with rigorous multiple comparison correction. The per-layer isolation methodology and BH step-up FDR correction establish a replicable framework for future expert quality analysis across model families.

3 Methodology

3.1 Model and Domains

We analyze Phi-mini-MoE, a 16-expert, 32-layer MoE model with 4096 hidden dimension and 65536 FFN intermediate dimension. Four domains were evaluated:

- **Math**: GSM8K-style arithmetic and algebra problems
- **Science**: Biology, chemistry, physics questions
- **General Knowledge**: Factual questions across diverse topics
- **Reasoning**: Logical inference and problem-solving tasks

Current analysis uses $n = 10$ items per domain per layer (total 5120 evaluations). Version 4.0 with $n = 50$ per domain is in preparation.

3.2 Per-Layer Expert Isolation

The core methodology is per-layer expert isolation via hook-based routing override:

1. For each layer ℓ and expert e, insert a hook that forces the MoE router at layer ℓ to select expert e with probability 1.0.
2. All other layers maintain normal router outputs.
3. Evaluate model on held-out test items.
4. Record log-probability of target answer tokens (both average and first-token metrics).
5. Compare forced expert performance against baseline (normal routing).

This isolates the *marginal* contribution of expert e at layer ℓ, conditioned on optimal routing everywhere else. We do not claim this measures absolute expert quality, only quality relative to the baseline routing strategy.

3.3 Log-Probability Scoring

We score each expert using log-probability of target answer tokens:

$$\text{Score} = \frac{1}{T} \sum_{t=1}^{T} \log P(y_t | x_{1:t-1}, \text{model}) \tag{1}$$

where $y_1, \ldots, y_T$ are target tokens and T is the number of target tokens. We also compute a length-invariant first-token metric to avoid biasing toward short answers.

Tokenization: Following prior work, we tokenize the prompt and target separately, then concatenate token IDs before forward pass. This prevents BPE boundary artifacts that can merge target content with the prompt.

3.4 Statistical Framework

3.4.1 Pairwise Comparisons: Wilcoxon Signed-Rank Test

For each expert-layer-domain combination, we perform a paired comparison of forced expert vs. baseline using the Wilcoxon signed-rank test (non-parametric, robust to non-normality):

$$H_0 : \text{Median}(\Delta \log\text{-prob}) = 0 \tag{2}$$

We report p-values and 95% confidence intervals on the median difference.

3.4.2 Multiple Comparison Correction: Benjamini-Hochberg Step-Up

With 896 tests (16 experts $\times$ 7 layers $\times$ 4 domains in core analysis), we apply Benjamini-Hochberg FDR correction at $\alpha = 0.05$:

1. Sort all p-values: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$.
2. Find largest k such that $p_{(k)} \leq \frac{k}{m}\alpha$.
3. Reject all hypotheses with rank $\leq k$.

This controls the false discovery rate (expected proportion of false positives among rejections) at level α.

Methodological Note: A common implementation error uses step-down logic (break on first failure), which is correct for Holm FWER control but incorrect for BH FDR. This error can qualitatively reverse conclusions. We verify our step-up implementation against statistical software.

3.4.3 Router Calibration: Spearman Correlation with Fisher Z-Transform

To measure whether the learned router's natural activation probabilities correlate with measured expert quality, we compute Spearman rank correlation (ρ) between:

- **X-axis:** Natural router activation probability for each expert at each layer (measured via hook instrumentation)
- **Y-axis:** Forced log-probability delta (expert quality from per-layer isolation)

Since we report ρ across multiple layers, we apply Fisher z-transform to enable proper averaging:

$$z = \operatorname{arctanh}(\rho) = \frac{1}{2} \ln \left(\frac{1+\rho}{1-\rho} \right) \tag{3}$$

Average across layers: $\bar{z} = \frac{1}{L} \sum_{\ell} z_{\ell}$
Transform back: $\bar{\rho} = \tanh(\bar{z})$
We clip estimates to $[-0.9999, +0.9999]$ to prevent numerical issues.

4 Results

4.1 Finding 1: Experts Have Measurable Domain Specialization

Table 1 shows the highest-performing expert for each domain, along with statistical confidence:

Table 1: Best expert per domain with confidence intervals and significance

Domain	Best Expert	Layer	Delta (log-prob)	95% CI	p-value	BH-FDR
Math	E7	L30	+0.684	[+0.358, +1.010]	0.002	**Sig.**
Science	E1	L3	+1.748	[+0.345, +3.151]	0.010	**Sig.**
General	E11	L4	+1.323	[+0.605, +2.042]	0.002	**Sig.**
Reasoning	E7	L30	+0.403	[+0.137, +0.668]	0.002	**Sig.**

Across all 896 expert-layer-domain combinations tested, 207 (23.1%) survive Benjamini-Hochberg FDR correction at $\alpha = 0.05$. The four domain-best experts shown above all survive correction, indicating statistically significant specialization.

Notably, the best experts are *different* across domains (E7 for math vs. E1 for science) and often at different layers (L30 vs. L3). This is not noise.

4.2 Finding 2: The Learned Router Is Miscalibrated

Table 2 reports Spearman rank correlation between natural router activation probability and forced expert log-probability delta:

Table 2: Router calibration: Spearman ρ between routing probability and expert quality

Domain	Fisher z-avg ρ	Sig. Layers	Interpretation
Math	+0.043	1/14	Effectively zero
Science	-0.027	2/14	Effectively zero
General	-0.146	1/14	Weak negative
Reasoning	+0.063	0/14	Effectively zero
Mean across domains	**-0.017**		**No correlation**

A well-calibrated router would show strong positive ρ (routes more to better experts). Instead we observe correlations indistinguishable from zero. For general knowledge, the correlation is actually *negative* ($\rho = -0.146$), indicating the router *preferentially avoids* the expert that helps most.

Figure **??** (when generated) will show scatter plots of natural activation vs. forced delta for representative layers, illustrating the lack of correlation visually.

4.3 Finding 3: A Single Expert Dominates All Domains

Table 3 shows natural router activation for the highest-activated expert across domains:

Table 3: Expert E2 dominance: Aggregated natural activation across all domains

Domain	Top Natural Expert	Aggregated Activation
Math	E2	0.0803
Science	E2	0.0718
General	E2	0.0734
Reasoning	E2	0.0724

With 16 experts, uniform routing assigns $1/16 = 0.0625$ to each. E2's average activation of ~ 0.075 is roughly 20% above uniform. While this is not a severe collapse, it is *systematic and misaligned with expert quality*: E2 never appears as the best expert for any domain (best experts are E7, E1, E11, E7).

This suggests the router has learned a domain-invariant preference for E2, perhaps because E2 works well in combination with typical experts at other layers, even though it is not optimal in isolation.

4.4 Finding 4: Layer Position Determines Expert Impact

Table 4 shows average maximum expert delta by layer position:

Table 4: Layer position effects: Expert quality varies dramatically by layer

Layer	Avg Best Delta	Role
L30	+0.779	Highest impact (near-output)
L3	+0.750	High impact (early)
L5	+0.715	High impact (early)
L4	+0.647	High impact (early)
L31	+0.441	Moderate (final)
L13	+0.066	Minimal (middle)
L8	-0.009	No effect (middle)

Expert identity matters most at early layers (L3-L5) and near-output layers (L30). Middle layers (L8, L13) show minimal expert differentiation. This suggests a **two-regime model**: early layers establish domain-specific representation, late layers refine, and middle layers perform domain-invariant computation where expert identity contributes little.

4.5 Finding 5: Semantic Routing as a Zero-Training Alternative

Expert profiling via gate activation fingerprints reveals domain specialization signatures. A cosine-similarity semantic router constructed from these profiles can route to appropriate experts

without any training. We demonstrate this in EXP-T18, though we emphasize this is proof-of-concept: **no downstream generation quality has been evaluated**. Whether semantic routing produces better completions than learned routing remains open.

4.6 Finding 6: Expert FFNs Can Be Grafted Across Models

EXP-T19 demonstrates that FFN modules from specialist models can be mechanically grafted into Phi-mini-MoE when dimensions align. Output statistics differ from native experts (as expected from specialists), but shapes are correct and outputs are non-degenerate. However, **no quality evaluation has been performed** on downstream task performance after grafting.

5 Methodological Contribution: The BH Step-Up Implementation

During development, we discovered a consequential bug in Benjamini-Hochberg implementation that qualitatively reverses conclusions:

Table 5: BH procedure variant comparison: why implementation matters

BH Variant	Logic	Surviving Tests	Conclusion
Step-down (incorrect)	Break on first $p \geq k/m \cdot \alpha$	0/896	No specialization
Step-up (correct)	Find largest k where $p_{(k)} \leq k/m \cdot \alpha$	**207/896**	**Real specialization**

The step-down variant (using `break` when a threshold is exceeded) is common in tutorials but incorrect for FDR control. It is correct for Holm's FWER procedure, but the original Benjamini & Hochberg (1995) paper specifies step-up: find the *largest* rank k satisfying the threshold, then reject all hypotheses $\leq k$.

With our $n = 10$ items, Wilcoxon p-values cluster around 0.002-0.01. Step-down breaks immediately when an early test fails its strict early threshold, even though hundreds of later tests pass their progressively lenient thresholds (e.g., $p_{(800)} \leq 800/896 \times 0.05 = 0.0446$). This is not an academic subtlety — it reversed our conclusions from "no evidence of specialization" to "robust evidence in 23.1% of tests."

We recommend practitioners always implement step-up and verify against reference implementations (e.g., scipy, R).

6 Discussion

6.1 Why Do Routers Miscalibrate?

Several hypotheses warrant investigation:

1. **Training objective mismatch:** Standard MoE training optimizes load balancing and task loss. Load-balancing auxiliary losses explicitly penalize non-uniform routing, which may prevent the router from learning quality-based specialization.

2. **Compositional optimization:** The router may optimize for expert *combinations* across layers, not individual expert quality at isolation. A router that performs well in composition could appear miscalibrated when tested in isolation.

3. **Attribution confusion:** The router may be learning input representations that select experts, but input-to-expert mapping may not correlate with expert quality (non-linear attribution).

4. **Insufficient gradient signal:** During training, gradients from all experts flow backward through the router. If one expert (E2) consistently receives high initial activation, it may dominate before quality-aware gradients accumulate.

Our per-layer isolation methodology cannot distinguish these hypotheses. Each requires different investigation.

6.2 Implications for MoE Architecture

1. **Per-layer routing flexibility:** Layer position dramatically affects expert impact (Table 4). MoE designs that allow per-layer routing strategies (as in DeepSeekMoE) may better exploit discovered specialization.

2. **Quality-aware training:** Standard MoE training does not directly optimize routing toward measured expert quality. Incorporating expert quality metrics as auxiliary losses or training signals could align routers with specialization.

3. **Expert assembly:** The existence of measurable specialization (Finding 1) suggests that selectively assembling experts for specific domains (semantic routing, expert grafting) could improve task-specific performance, even without retraining.

6.3 Limitations

We state the following limitations transparently:

1. **Single model:** All results are from Phi-mini-MoE. Generalization to Mixtral, DeepSeek-MoE, OLMoE, and other MoE models is unknown.

2. **Test set size:** v3.2 uses $n = 10$ items per domain per layer (total 5120 evaluations). Version 4.0 with $n = 50$ is in preparation. Smaller n increases statistical noise and reduces BH power.

3. **Marginal vs. compositional quality:** Per-layer isolation measures marginal contribution of each expert conditioned on normal routing elsewhere. The router may optimize for expert combinations across layers rather than individual quality. Our methodology cannot distinguish miscalibration (bad for isolation) from compositional optimization (good in context). The Spearman $\rho \approx 0$ finding applies specifically to the marginal/isolation sense.

4. **No end-to-end generation quality:** We measure log-probability deltas, not downstream task performance. Semantic routing and expert grafting are unvalidated on real tasks.

5. **No random routing control in v3.2:** v4.0 adds a random expert baseline to establish the floor for grade deltas. Without this, we cannot decompose learned routing quality into routing vs. expert averaging effects.

6. **Phi-mini-MoE quality:** The base model has known quality issues (incorrect answers at baseline). Findings about expert specialization are relative to baseline, not absolute assessments of expert ability.

7. **E2 dominance magnitude:** E2's preference (0.075) is only ∼20% above uniform (0.0625), not a severe collapse. The finding is systematic misalignment with quality, not extreme concentration.

7 Future Work

1. **Scale test sets:** Expand to $n \geq 50$ per domain to increase BH statistical power and reduce noise.
2. **Multi-model validation:** Replicate analysis on Mixtral-8x7B, DeepSeek-MoE, and OL-MoE to establish whether miscalibration generalizes.
3. **End-to-end generation quality:** Evaluate semantic routing and expert grafting on downstream benchmarks (MMLU, GSM8K, etc.).
4. **Random routing baseline:** Add uniform random expert selection to establish the floor for expert quality deltas.
5. **Training dynamics:** Analyze how router calibration evolves during training. Does miscalibration arise early or develop over training?
6. **Quality-aware auxiliary loss:** Explore training MoE models with auxiliary losses that directly optimize for expert quality correlation.

8 Conclusion

We provide statistical evidence that learned routers in Mixture-of-Experts models are miscalibrated with respect to expert quality. Experts exhibit significant domain specialization (207/896 tests survive FDR correction), yet the learned router shows near-zero correlation ($\rho = -0.017$) between routing probability and expert quality. A single mediocre expert (E2) is preferentially routed to across all domains, despite never being optimal.

These findings do not imply that MoE routing is useless. The router may optimize for expert combinations across layers, or load balancing may be the dominant training signal. However, they reveal an important gap between assumed behavior (quality-aware routing) and observed behavior (quality-indifferent routing), opening new research directions for MoE architecture design and training.

The per-layer expert isolation methodology and Benjamini-Hochberg step-up FDR framework establish a replicable foundation for future expert analysis across model families. We make experiment code and results available to the community.

References

[1] Fedus, W., Lepikhin, D., et al. (2022). Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research*, 23(120), 1-39.

[2] Zhou, Y., Lei, T., et al. (2022). Experts Choose: A Flexible Framework for Mixture of Experts. In *ICML 2022*.

[3] Dai, S., Wang, Y., et al. (2024). DeepSeekMoE: Fine-Grained Expert Segmentation for Expert-Shared Models. In *ICLR 2024*.

[4] Lepikhin, D., Lee, H., et al. (2020). GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. In *ICLR 2021*.

[5] Shazeer, N., Mirhoseini, A., et al. (2017). Outrageously Large Neural Networks for Efficient Conditional Computation. *arXiv preprint arXiv:1701.06538*.

[6] Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society Series B*, 57(1), 289-300.

[7] Lepikhin, D., Lee, H., & Zhou, Y. (2020). GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. In *ICLR 2021*.

A Implementation Details

A.1 Single Layer Expert Hook

The SingleLayerExpertHook registers with each MoE layer to intercept router outputs:

```
class SingleLayerExpertHook:
    def __init__(self, target_expert):
        self.target_expert = target_expert

    def __call__(self, router_outputs):
        # Force router to select target_expert with p=1.0
        batch_size = router_outputs.shape[0]
        forced = torch.zeros_like(router_outputs)
        forced[:, self.target_expert] = 1.0
        return forced
```

For each evaluation, we register exactly one hook at the target layer, leaving all other layers with normal routing.

A.2 Reproducibility

All experiments use fixed random seeds: `SEED=42` for PyTorch, NumPy, and Python random. Model loading uses `device_map='auto'` with HuggingFace transformers 4.36.0. Results are serialized with JSON including:

- Timestamp and git commit hash
- Model name, device, CUDA version
- Test set checksums for reproducibility
- Per-layer expert scores and router activations

A.3 Statistical Software Verification

BH step-up FDR correction was verified against:

- `scipy.stats.false_discovery_control(method='fdr_bh')`
- R `p.adjust(..., method='BH')`

Spearman correlation and Fisher z-transform verified against `scipy.stats.spearmanr` and manual arctanh/tanh implementation.

B Complete Results Tables

[Full detailed tables of all 896 expert-layer-domain results are available in the supplementary results/ directory]