Who Routes the Router: Rethinking the Evaluation of LLM Routing Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

The proliferation of Large Language Models (LLMs), each with different capabilities and costs, has driven the need for LLM routers that intelligently and dynamically select the best model for a given query. Evaluating these routing systems is important yet inherently challenging due to the complex interplay of multiple factors: the selection of representative input queries, the composition of the model pool, and the definition of comprehensive evaluation metrics for optimal routing decisions. Through extensive analysis of existing benchmarks, we identify critical limitations that may lead to incomplete results and/or misleading conclusions about router performance: (1) limited task diversity, (2) imbalanced model pools, and (3) oversimplified evaluation methodologies. To address these limitations, we propose a novel evaluation framework that incorporates diverse task distributions, a balanced model pool with complementary model strengths, and multi-faceted metrics that reflect real-world deployment scenarios. We implement this framework as an open-source benchmark, the code and dataset are shared anonymously at: `https://anonymous.4open.science/r/rethinking-routing-evaluation-DE30`

1 Introduction

The rapid proliferation of Large Language Models (LLMs) presents a critical challenge: *which model best achieves desired performance while minimizing cost?* [16, 3, 9] Models like GPT-o3 excel at complex reasoning but cost significantly more than alternatives like Mixtral-8×7B [6], while domain-specialized models often outperform general ones in their expertise areas [18, 15].

LLM routing systems address this by dynamically selecting the most appropriate model for each query [3, 16, 9]. A routing system maps queries $p \in \mathcal{P}$ to models $m \in \mathcal{M}$ via function $R : \mathcal{P} \to \mathcal{M}$, optimizing objectives like: $m^* = \arg\max_{m_i \in \mathcal{M}} \hat{q}_i(p), \quad m^* = \arg\min_{m_i \in \mathcal{M}} c_i(p)$ subject to $\hat{q}_i(p) \geq T$

To design effective LLM routers, **rigorous evaluation becomes especially crucial** [4, 5]. However, evaluating LLM routers is inherently challenging, as optimal routing decisions are context-dependent and shaped by specific priorities and constraints. Unlike evaluating LLM performance where each query can be assessed against ground truth, optimal routing strategies depend on specific deployment contexts. Even for the same query, the optimal routing decision may vary under different constraints.

Our analysis reveals current benchmarks suffer from: (1) limited task diversity, (2) imbalanced model pools where one model dominates, and (3) oversimplified metrics focusing only on accuracy. We introduce RouterBench+, featuring 33,337 queries across 68 categories, 85 models with complementary strengths, and multi-faceted evaluation with OOD testing. Our contributions include:

- Systematic analysis revealing critical limitations in existing routing benchmarks
- Evaluation methodology addressing task diversity, model balance, and realistic metrics
- Open-source platform enabling rigorous routing assessment under realistic conditions

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

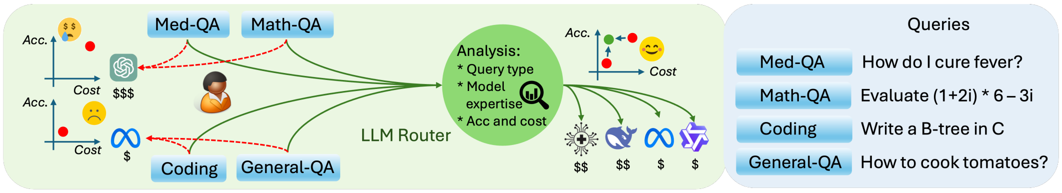


Figure 1: An illustration of LLM routing systems. An ideal LLM router should choose the model with highest expected performance under the specified constraints like costs.

2 Rethinking Current Evaluation Practices

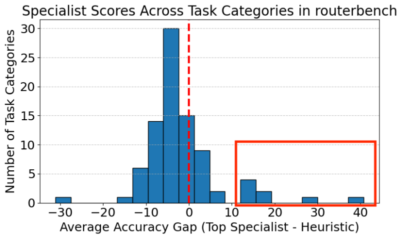
We analyze current routing evaluation practices across three dimensions: tasks, models, and evaluation metrics, identifying critical limitations through extensive experimentation. We use three prominent benchmarks: EmbedLLM [22], RouterBench [4], and MixInstruct [7]. We test both clustering-based (K-Means, K-NN) and learning-based (MLP, Matrix Factorization) routing methods, along with Heuristic and Oracle baselines. Routing quality is visualized using a *deferral curve* that captures the trade-off between routing quality and resource usage under cost constraints. For details of the benchmarks, routing methods, and evaluation metrics, please refer to Appendix B–E.

2.1 Tasks: More Diversity and Less Redundancy

Problem 1: Lack of Specialized Tasks. Generally, tasks can be categorized as **common-sense** tasks where general models perform well (e.g., `piqa` with 78% average accuracy), and **domain-specific** tasks where specialists excel (e.g., `medmcqa` where specialists achieve 69.8% vs 41.73% for general models). However, current benchmarks are biased toward common-sense tasks, which fails to evaluate routers' ability to handle domain-specific tasks that benefit most from model routing.

To quantify this imbalance, we propose a **specialist score** for each task: $\text{specialist\_score}_{\text{task}} = \mathbb{E}_{b\in\mathcal{B}}[\max_{m\in\mathcal{M}^{(b)}_{\text{non-gen}}} \text{ACC}^{(b)}_{m,t} - \text{ACC}^{(b)}_{\text{gen},t}]$, measuring the average performance gap between the best specialist and generalist models across cost budgets. Figure 2(a) shows RouterBench lacks sufficient specialist tasks, with EmbedLLM showing similar patterns (see Appendix F).

Problem 2: Task Redundancy. Current benchmarks also suffer from significant task redundancy. Through cosine similarity analysis (detailed in Appendix F), we identified 1,346 duplicate query groups where 99.9% contained label disagreements across models—far exceeding the overall label mismatch rate of 37.7%. Removing such duplicates improved performance for learning-based methods (Figure 2(b)), confirming that duplicate queries with conflicting labels mislead routers.



(a) Limited specialized tasks

Method	Avg Acc (%)		Peak Acc (%)	
	Orig	Clean	Orig	Clean
K-NN	**54.35**	54.04	**67.37**	66.50
KMeans	**54.03**	54.00	**66.77**	66.70
MLP	53.78	**53.84**	64.17	**65.13**
MF	50.48	**50.90**	60.07	**60.87**

(b) Performance after removing duplicates

Figure 2: Task diversity issues in current benchmarks: (a) specialist scores reveal limited specialized tasks; (b) removing duplicate queries improves learning-based methods.

Insights ❶. Current benchmarks overestimate the value of large but non-diverse training sets; in reality, much of the routing signal is concentrated in a smaller, more representative subset of tasks. To build more effective routing benchmarks, we should improve task diversity—especially by including more domain-specific tasks—and reduce redundancy, particularly tasks with inconsistent labels.

2.2 Models: More Specialists and Less Dominance

Problem 1: Model Dominance. We quantify model dominance using *average rank* across tasks. RouterBench's top model ranks 1.36 on average (near-universal best performance), while EmbedLLM's top model ranks 6.4 (more balanced competition) (see Appendix G). Without specialist models, routers could simply select the best generalist, which make advanced routing unnecessary.

Effective Expert Model Extension. To address this limitation, we propose augmenting the model pool with *pseudo-specialist models*—artificial models designed to perform well on specific tasks and average elsewhere. These pseudo-models are not meant for deployment but serve as controlled interventions to examine how task-specialized models influence routing behavior. They allow us to test whether the router moves beyond favoring top generalists and begins making more diverse, task-aware selections.

For each pseudo-specialist model, we need to identify appropriate task types where they can demonstrate their expertise. We selected three specific task types to create our pseudo-specialist models, each designed to excel at one particular task. These tasks satisfy the following criteria:

- The mean accuracy across existing models is low, suggesting that they are challenging.
- The gap between the best general model and the mean is modest, so no existing model dominates.
- The task has a non-negligible representation that impacts the overall performance in the benchmark.

Each pseudo model is injected into the benchmark with high performance on a specific task (Table 1) and average performance elsewhere, simulating models trained for niche tasks. We use EMBEDLLM as the task selection pool.

Table 1: Selected tasks for pseudo specialist models.

Task	Prompt %	Mean Acc.	Best Model Acc.	Pseudo Model Acc.
Social Reasoning	5.42	33.76%	36.22%	**65.00%**
Logical Reasoning	1.82	28.28%	45.93%	**70.00%**
Graduatel-Lvl Reasoning	3.23	22.44%	33.51%	**60.00%**

We further define the *agreement score* as the average percentage of queries for which a router selects the same model as the heuristic router. This metric reflects how closely a learned router mimics static generalist selection. A lower score indicates more diverse, task-specific choices, suggesting less reliance on the generalist strategy. As shown in Table 2, overall agreement with the heuristic router drops slightly across all methods. However, on the tasks targeted by the pseudo models, the reduction is significantly more pronounced.

Table 2: Changes in router agreement with the top-1 generalist model after adding pseudo specialist models. Negative values indicate decreased reliance on the dominant model.

Task	K-NN	KMeans	MF	MLP
Overall	-0.84	-2.40	-0.64	**-8.40**
logiqa	**-20.55**	**-31.03**	-2.81	**-17.48**
social_iqa	-2.69	0.00	+0.29	**-7.92**
gpqa	-1.59	**-13.15**	+1.90	**-9.75**

Problem 2: Model Redundancy. We also observe redundancy in the model pool. Using a Jaccard-style similarity score (detailed in Appendix G), we reduced EmbedLLM's model pool from 112 to 82 (27% reduction) without degrading routing performance, confirming that meaningful routing decisions can be made with a leaner model pool.

Insights ❷. Effective routing evaluation depends on a model pool with *meaningful* diversity, both in capability and specialization. Rather than including many models with overlapping strengths, the pool should consist of models with distinct specialties. A simple yet effective way to enhance current model pools is to introduce pseudo-specialist models that simulate task-specific expertise, encouraging routers to move beyond generic selection and make more nuanced, task-aware decisions.

2.3 Evaluation Paradigms: Comprehensive Measurements

Problems. Current evaluation paradigms suffer from at least two limitations: (1) *Limited cost awareness*: Existing evaluations often overlook the importance of selecting smaller, more efficient models when appropriate, leading to inflated costs and suboptimal routing decisions. (2) *Lack of OOD evaluation*: Current frameworks rarely test router performance on OOD inputs, an essential aspect for ensuring robustness in real-world deployments.

Solution 1: **Cost-Aware Routing Evaluation ❸.** We introduce a paradigm to assess how routers balance between strong generalist and lightweight models (Figure 3), producing trade-off curves between accuracy and model cost (additional results in Appendix H.1).

Solution 2: **OOD Testing ❹.** We evaluate robustness by holding out entire task categories (e.g., math tasks) from training. Table 3 shows significant performance drops, revealing brittleness of current approaches.

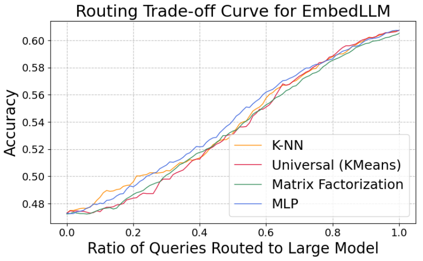


Figure 3: Binary routing evaluation: Llama-2 7B vs 70B trade-offs.

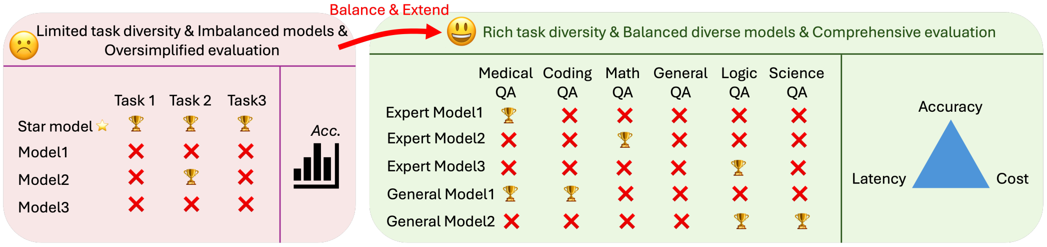


Figure 4: The improvement of our proposed benchmark.

Table 3: OOD performance degradation on math categories.

Category	K-NN Δ	KMeans Δ	MF Δ	MLP Δ
mathqa	-9.29	-16.88	-6.33	-14.34
asdiv	-58.59	-69.19	-40.40	-57.07
gsm8k	-14.28	-14.29	-29.47	-35.72

Insights. Current benchmarks inadequately assess router performance in realistic scenarios. The OOD performance degradation (Table 3) reveals the brittleness of existing approaches with novel queries, highlighting the need for better generalization testing. Additionally, the binary routing paradigm (Figure 3) shows that routing algorithms have distinct efficiency-performance trade-offs, requiring evaluation beyond single-point metrics.

3 Remastered Benchmark Design

Figure 4 shows the strength of our evaluation framework, we address the identified limitations through three key design principles:

Diverse task distributions ❶: We subsample tasks from EmbedLLM using the proposed *specialist score*, emphasizing tasks where non-generalist models provide additional value. This creates a task pool with both broad coverage and meaningful routing opportunities.

Balanced model pool ❷: We eliminate redundancy using similarity-aware greedy pruning (reducing 30 models) and introduce three pseudo-specialist models for challenging tasks, ensuring no single model dominates across all tasks.

Multi-faceted evaluation ❸❹: We combine classification-based and routing-rate paradigms with explicit OOD testing, capturing cost-performance trade-offs and real-world deployment readiness.

The final dataset contains 85 models, 68 categories, and 33,337 queries (3 million datapoints).

4 Results and Discussion

Benchmark Performance. We evaluate routing methods on our remastered benchmark. K-NN achieves the highest performance with an area under the deferral curve of 0.567 and peak accuracy of 69.83% (Table 4). Our dataset successfully mitigates single model dominance as shown by the deferral curves (Figure 11 in Appendix). Binary routing results reveal distinct efficiency-performance patterns across model combinations (see Figure 12 in Appendix).

Table 4: Area and peak accuracy of routing methods.

Method	Area ↑	Peak (%) ↑
K-NN	0.567	69.83
KMeans	0.560	68.93
MLP	0.554	67.60
MF	0.515	61.60
Heuristic	0.507	60.73

Key Findings. Our analysis reveals three critical limitations in current routing evaluation: (1) benchmarks lack task diversity, particularly domain-specific tasks where specialists excel; (2) model pools suffer from single-model dominance, making routing trivial; and (3) evaluation methodologies ignore cost-performance trade-offs and OOD robustness. These findings explain why existing routers often perform only marginally better than simple heuristics.

Implications. Effective routing evaluation requires careful attention to all three components—tasks, models, and metrics. Our specialist score and pseudo-specialist models provide practical tools for constructing balanced benchmarks. The multi-faceted evaluation approach, combining traditional metrics with binary routing paradigms and OOD testing, offers comprehensive insights into router behavior.

Future Directions. This work establishes a foundation for more rigorous LLM routing evaluation. As the LLM ecosystem continues to evolve, maintaining diverse, balanced benchmarks will remain crucial. Future work should explore automated benchmark construction and dynamic adaptation to emerging models and tasks.

References

[1] Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models, 2024.

[2] Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for llms, 2025.

[3] Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections, 2025.

[4] Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system, 2024.

[5] Zhongzhan Huang, Guoming Ling, Vincent S Liang, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. *arXiv preprint arXiv:2503.10657*, 2025.

[6] 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.

[7] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

[8] Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Zifeng Wang, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient llm inference, 2025.

[9] Yang Li. Llm bandit: Cost-efficient llm generation via preference-conditioned dynamic routing, 2025.

[10] Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1964–1974, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[11] Sentence-Transformers. all-minilm-l12-v2. `https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2`, 2021. Accessed: 2025-05-13.

[12] Sentence-Transformers. all-mpnet-base-v2. `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`, 2021. Accessed: 2025-05-13.

[13] KV Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. Harnessing the power of multiple minds: Lessons learned from llm routing. *arXiv preprint arXiv:2405.00467*, 2024.

[14] Dimitris Stripelis, Zijian Hu, Jipeng Zhang, Zhaozhuo Xu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Salman Avestimehr, and Chaoyang He. Tensoropera router: A multi-model router for efficient llm inference. *arXiv preprint arXiv:2408.12320*, 2024.

[15] Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Charles Lau, Ryutaro Tanno, Ira Ktena, et al. Towards generalist biomedical ai. *Nejm Ai*, 1(3):AIoa2300138, 2024.

[16] Clovis Varangot-Reille, Christophe Bouvard, Antoine Gourru, Mathieu Ciancone, Marion Schaeffer, and François Jacquenet. Doing more with less – implementing routing strategies in large language model-based systems: An extended survey, 2025.

[17] Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. Mixllm: Dynamic routing in mixed large language models. *arXiv preprint arXiv:2502.18482*, 2025.

[18] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[19] Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. Masrouter: Learning to route llms for multi-agent systems, 2025.

[20] Tuo Zhang, Asal Mehradfar, Dimitrios Dimitriadis, and Salman Avestimehr. Leveraging uncertainty estimation for efficient llm routing, 2025.

[21] Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Capability instruction tuning: A new paradigm for dynamic llm routing, 2025.

[22] Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. Embedllm: Learning compact representations of large language models. *arXiv preprint arXiv:2410.02223*, 2024.

A Related Work

LLM Model Selection and Routing. Intelligent LLM routers have emerged to route queries across diverse models to balance performance, cost, and latency [3, 16, 9, 19, 20]. Routing strategies can be categorized as predictive and non-predictive [16, 4]. Predictive approaches include classification based on prompt features [13], graph-based methods (GraphRouter [3]), dynamic routing (MixLLM [17]), and multi-armed bandit formulations (LLM Bandit [9]). Non-predictive methods include cascading, while hybrid approaches like Cascade Routing [2] combine routing flexibility with sequential processing. Frameworks like TensorOpera Router [14] further enhance multi-model inference efficiency. The proliferation of LLM routing methods has produced the requirement for effective router evaluation [1, 10, 21].

Benchmarks for Multi-LLM Systems. Several benchmarks have been developed to evaluate routing strategies. RouterBench [4] provides a framework with inference outcomes across models and tasks [2, 17]. EmbedLLM [22] introduces compact vector embeddings for efficient model selection. MixInstruct [7] offers a mixture-of-instructions dataset with a two-stage ensembling approach. RouterEval [5] presents a large-scale benchmark with over 8,500 models and 200 million routing records. These benchmarks are crucial for developing robust routing systems that enable efficient, cost-effective LLM deployment [3, 13, 16, 9].

Despite the growing body of work on LLM routing techniques and benchmarks, we identify a critical gap: **the evaluation methodology itself has not been systematically examined**. Even the most comprehensive and recently released benchmarks, such as RouterEval [5], primarily aggregate large volumes of data and models without addressing fundamental flaws in evaluation design. This paper fills that gap by critically analyzing current evaluation practices and providing concrete recommendations for improvement.

B Details about Text Encoder

Text encoder is a critical component of LLM routers, which transforms input prompts into embeddings used for routing decisions. To ensure faithful and fair comparison, we follow prior work [22, 4] and adopt consistent encoder choices per benchmark: we use `all-MiniLM-L12-v2` [11] for ROUTERBENCH and MIXINSTRUCT, and `all-mpnet-base-v2` [12] for EMBEDLLM.

C Details about Benchmarks

Table 5 summarizes the statistics of used benchmarks. EmbedLLM provides the largest number of models, while RouterBench provides a realistic cost setting. **MixInstruct** focuses on open-domain user prompts, using soft metrics like BARTScore to evaluate output quality.

Table 5: Comparison of benchmark datasets for LLM routing evaluation.

Benchmark	# Models	# Queries	# Categories	Metric	Cost Info
EmbedLLM [22]	112	35,673	80	Binary (0/1)	param size (B)
RouterBench [4]	11	36,497	86	Binary (0/1)	USD per 1k queries
MixInstruct [7]	12	110,000	5 (Open-domain)	exp(BARTScore)	param size (B)

D Details about Routing Methods

The state-of-the-art LLM routing approaches fall into two primary categories: *clustering-based* and *learning-based*. We also include two *reference baselines* to contextualize performance.

- **K-Means** [8]: This method clusters training queries into K clusters based on their embeddings. Given a test query q, the router finds the closest cluster C_k and selects the model m^* that performs best on average within that cluster:

$$m^* = \arg\max_{m_i \in \mathcal{M}} \left[\frac{1}{|C_k|} \sum_{l \in C_k} \text{metric}(m_i, l) \right]$$

where C_k is the set of training prompts in the cluster of q, and metric denotes either a binary correctness label or $\exp(\text{BARTScore})$.

- **K-NN** [4]: Instead of relying on cluster centroids, this method finds the K nearest neighbors of the query q in the training set (based on embedding distance) and routes to the model with the highest average score on those neighbors.
- **MLP** [4]: For each LLM m_i, a separate MLP is trained to predict the performance score for query q:
$$P_i(x) = f(W_n \cdot \sigma(\ldots\sigma(W_1 \cdot x + b_1)\ldots) + b_n)$$
where x is the query embedding, σ denotes the activation function, and f is the final output layer. The model m^* with the highest predicted score $P_i(q)$ is selected.
- **Collaborative Filtering (Matrix Factorization)** [22]: This method treats the model routing task as a matrix completion problem. Given a binary matrix $Y \in \{0,1\}^{M\times Q}$ representing whether model m_i correctly answered query q_j, it learns latent embeddings for models and queries by factorizing Y as:
$$Y_{ij} \approx u_i^\top v_j$$
where $u_i \in \mathbb{R}^d$ is the latent embedding for model m_i and $v_j \in \mathbb{R}^d$ for query q_j. At inference time, the router computes v_q (e.g., via a linear projection from query embedding) and selects the model with the highest predicted score:
$$m^* = \arg\max_{m_i \in \mathcal{M}} u_i^\top v_q$$
- **Heuristic Router**: This baseline selects the best-performing model from the training set for each cost budget. At each test time cost step, it routes all queries to the model that achieved the highest average training accuracy within the allowed cost:
$$m^* = \arg\max_{m_i \in \mathcal{M},\ \text{cost}(m_i) \leq c} \text{TrainAcc}(m_i)$$
- **Oracle Router**: This upper-bound baseline assumes access to the ground truth performance of all models at test time. For each query, it routes to the best model among those allowed by the cost constraint:
$$m^* = \arg\max_{m_i \in \mathcal{M},\ \text{cost}(m_i) \leq c} \text{metric}(m_i, q)$$
It represents the best possible routing performance under the given budget.

E Details about Evaluation Metrics and Deferrel Curve

Evaluation Metric. We evaluate routing performance using metrics aligned with each benchmark's design. For RouterBench [4] and EmbedLLM [22], the correctness label is binary—each LLM either answers a query correctly or not. For MixInstruct [7], we adopt the exponentiated BARTScore, following prior work [8, 7]. While MixInstruct was originally intended to benchmark ensemble generation quality from outputs of multiple LLMs, recent works have adapted it for routing by assigning scores to individual LLM responses based on similarity to GPT-4. However, this introduces a dependency on GPT-4 as a reference model, which we will discuss further in Section 2.3.

Deferral Curve. Routing quality is visualized using a *deferral curve*, where the x-axis corresponds to the model cost budget and the y-axis reflects routing quality (accuracy or exp(BARTScore)). The cost budget represents the maximum cost (e.g., in dollars) a router can spend per query. However, because actual API pricing varies and is not always available, prior work [8] approximates cost using the number of model parameters—a practical proxy that correlates with both latency and financial cost for EmbedLLM [22] and MixInstruct [7]. This deferral curve captures the trade-off between routing quality and resource usage, allowing comparison of different routing strategies under cost constraints.

F Supplementary Result for Task Diversity

Here we provide more results and discussions on **Task Diversity Problems** in Section 2.1. Figure 5 shows the specialist scores for EMBEDLLM, complementing the ROUTERBENCH results shown in the main text. Both benchmarks exhibit a similar lack of specialist-demanding tasks.

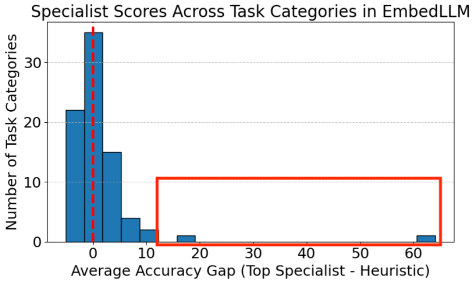


Figure 5: Specialist scores in EMBEDLLM dataset, showing limited specialized tasks similar to ROUTERBENCH.

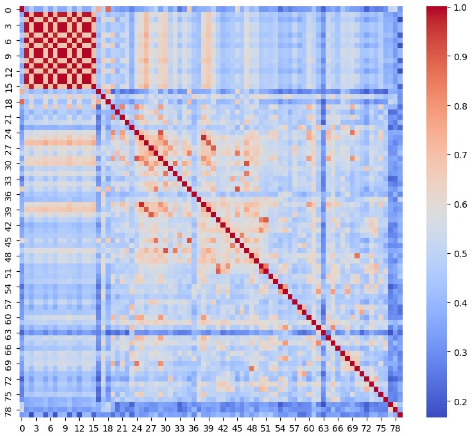


Figure 6: Category similarity heatmap based on average query embeddings. Redundancy is visible across GPQA-like categories (Upper-Left).

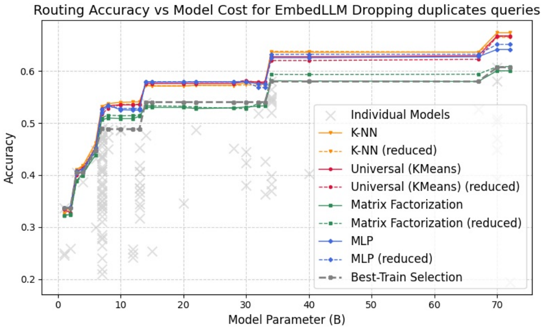


Figure 7: Routing accuracy when removing duplicate categories (e.g., GPQA variants). Performance is preserved even under OOD evaluation.

Additionally, as shown in Figure 6, several task categories exhibit high similarity in their average query embeddings. For instance, GPQA-like categories cluster tightly in the embedding space, suggesting that they may not offer distinct routing challenges.

In our experiments, we found that even after removing duplicate categories from the training set—those identified as redundant in the heatmap—the router still performs strongly. Figure 7 shows that this holds true even under OOD evaluation, where the dropped categories are tested at inference time. This suggests that current benchmarks may overestimate the value of large or diverse-looking training sets when, in reality, much of the routing signal is concentrated in a smaller, more representative subset of tasks. We also empirically assess the redundancy within categories, where we progressively dropped a portion of training data within each category and retrained the router.

G Supplementary Result for Model Diversity

Here we provide more results and discussions on **Model Dominance and Redundancy Problems** in Section 2.2.

G.1 Model Dominance Analysis

As detailed in the main text, we compute each model's average rank across task categories to quantify model dominance. Table 6 shows the full results. In ROUTERBENCH, model ID 5 dominates with an average rank of 1.36, meaning it is the best-performing model for most tasks. In contrast, EMBEDLLM's more diverse model pool shows less dominance, with the top model achieving only 6.43 average rank.

Table 6: Top-5 models by their average rank across tasks. Lower values indicate greater dominance.

EmbedLLM			RouterBench		
Rank	**Model ID**	**Avg. Rank (↓)**	**Rank**	**Model ID**	**Avg. Rank (↓)**
1	50	6.43	1	5	1.36
2	83	9.88	2	10	3.20
3	42	10.03	3	4	3.78
4	49	10.95	4	9	3.88
5	5	11.24	5	3	5.39

G.2 Model Redundancy Analysis

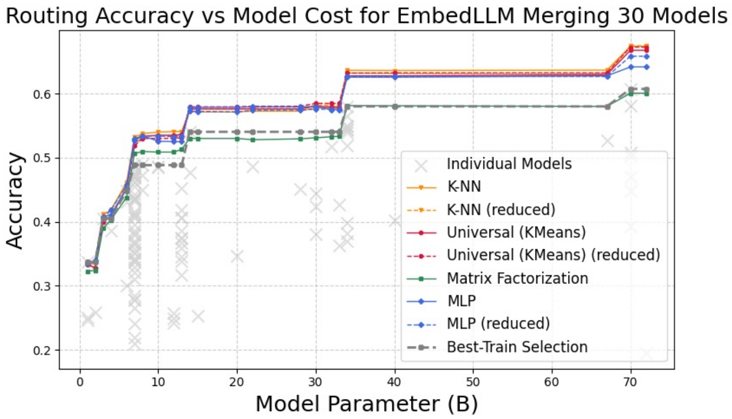


Figure 8: Performance comparison after reducing the model pool by 30 models. This shows that routers can maintain routing effectiveness across different cost budgets.

We observed redundancy in the model pool, as evidenced by overlapping performance points (Individual Models' grey crossings) across cost settings in Figure 8. Such redundancy adds little value for training or evaluating router performance. We quantify model-level similarity using a Jaccard-style score based on shared correct predictions:

$$\text{sim}(m_i, m_j) = \frac{|\{q \mid m_i(q) = 1 \wedge m_j(q) = 1\}|}{|\{q \mid m_i(q) = 1 \vee m_j(q) = 1\}|}$$

where $m_i(q)$ denotes whether model m_i answered query q correctly. This metric captures functional overlap across the entire benchmark.

To validate this, we propose a greedy pruning strategy to reduce model redundancy while preserving routing effectiveness. At each step, we compute a score for each model based on:

$$\text{score}(m_i) = \lambda \cdot \text{Accuracy}(m_i) - (1 - \lambda) \cdot \text{AvgSim}(m_i)$$

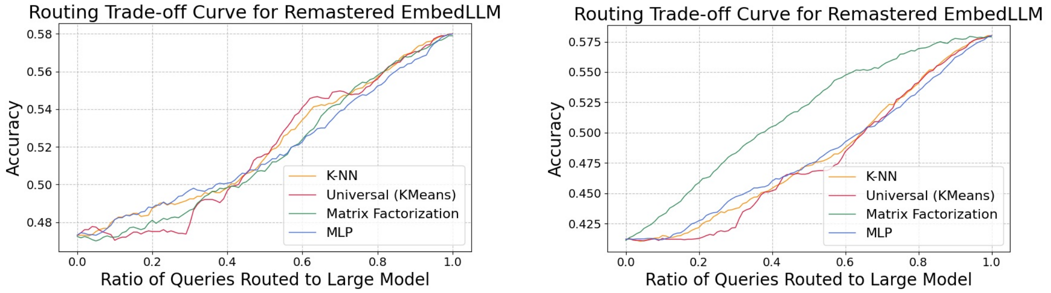
where AvgSim(m_i) is the average Jaccard similarity of model m_i to all other models (based on overlapping correct predictions), and λ balances performance versus uniqueness. The model with the lowest score is removed, and the process repeats until a target number of models remains.

We apply this strategy to the `EmbedLLM` benchmark, reducing the model pool from 112 to 82 (a 27% reduction). As shown in Figure 8, routing performance across methods remains comparable to the full model pool. This demonstrates that removing redundant models does not degrade routing quality and that meaningful routing decisions can still be made with a leaner model pool.

H Supplementary Result for Evaluation Methodology

H.1 Binary Routing Evaluation Details

To complement the traditional cost-accuracy deferral curves, we introduced a binary routing evaluation paradigm in Section 2.3 to assess how effectively a router balances between a strong generalist (large



(a) Llama-2 7B and CausalLM-34B-Beta

(b) Mistral 7B and CausalLM-34B-Beta

Figure 9: Binary routing evaluation paradigm showing performance trade-offs.

model) and a lightweight alternative (small model). Here, we provide additional details about the evaluation setup and key observations.

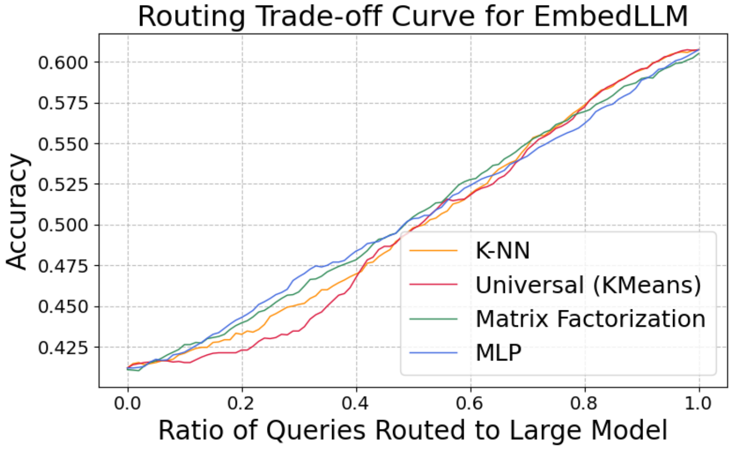


Figure 10: Binary routing evaluation: Mistral 7B vs Llama-2 70B trade-offs, complementing the Llama-2 7B vs 70B results shown in the main text.

We fix the large model to be `CausalLM-34B-Beta`, given its similar superior performance across a wide range of general-purpose tasks, comparable to that of 70B-sized models. For small models, we consider two widely used options: `Mistral-7b-v0.1` and `LLaMA-2-7b-chat-hf`. These models represent different trade-offs in model families and capability, making them ideal candidates for evaluating routing flexibility.

In this setting, each routing method ranks the queries by its confidence score for the small model and routes a varying fraction of queries accordingly, as in Figure 9. The remaining queries are deferred to the large model. This produces a continuous accuracy curve as a function of the fraction of queries routed to the large model.

Across both small model settings, we observe that learned routers generally follow a linear trade-off curve, indicating that they lack precise mechanisms to identify which queries can be reliably handled by the small model. Notably, clustering-based methods perform sub-linearly at lower deferral ratios, suggesting they often misclassify harder queries as easy ones and route them to the small models. This reinforces the need for more fine-grained routing strategies that can better distinguish between simple and complex inputs. Surprisingly, Matrix Factorization performed extremely well on classifying between Mistral-7B and CausalLM-34B-Beta, suggesting the potential of learning-based methods in certain model pair settings.

Table 7: OOD Performance change on selected categories in EMBEDLLM when these categories are excluded from training.

Category	K-NN Δ	KMeans Δ	MF Δ	MLP Δ
`mathqa`	-9.29	-16.88	-6.33	-14.34
`asdiv`	-58.59	-69.19	-40.40	-57.07
`gsm8k`	-14.28	-14.29	-29.47	-35.72
`medmcqa`	-11.58	-7.91	-6.78	-9.89
`mmlu_clinical_knowledge`	0.00	+7.41	-14.82	-3.70
Average	-18.75	-20.17	-19.56	-24.14

H.2 OOD Routing Evaluation Details

We evaluate the robustness of routing methods under out-of-distribution (OOD) scenarios by training and evaluating routers on different domains. We consider two distinct OOD settings: (1) excluding all math-related queries (e.g., `mathqa`, `asdiv`, `gsm8k`), and (2) excluding all medical-related queries (e.g., `medmcqa`, `mmlu_clinical_knowledge`). These categories are chosen for their semantic distinctiveness and task specificity, providing strong settings to evaluate how well routers generalize to unseen topics.

As shown in Table 7, all routing methods suffer performance degradation in OOD settings, with the most significant drops occurring on `asdiv` and `gsm8k`. MLP-based routers tend to experience the steepest accuracy declines overall, while matrix factorization (MF) demonstrates greater robustness, particularly on math-related tasks.

These results highlight that existing routing strategies are brittle when deployed in domains unseen during training, reinforcing the need for more semantically aware or domain-adaptive routing mechanisms.

H.3 Results on Remastered Benchmark

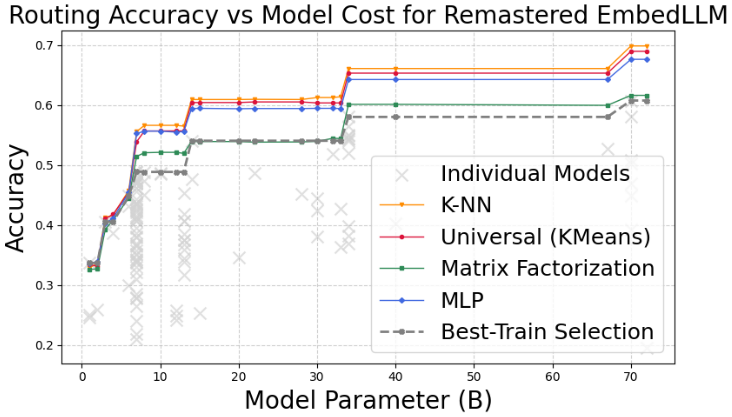
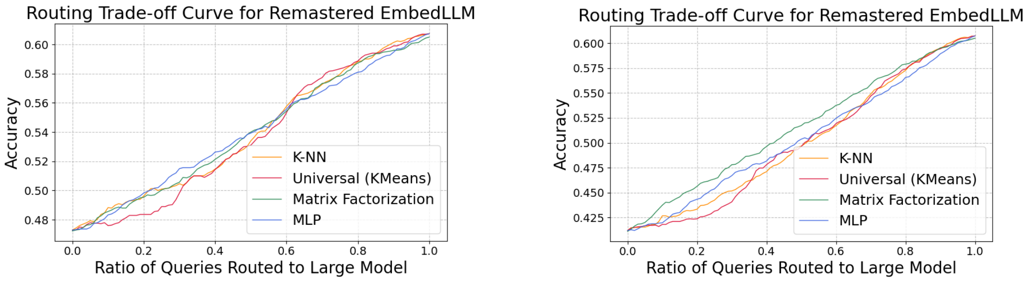


Figure 11: Deferral curves on our Remastered Benchmark showing routing performance across different cost budgets.

I Supplementary Result on MIX-INSTRUCT

In Figure 13, we present the routing results in deferral curve on MIX-INSTRUCT dataset. While the same baseline routers are evaluated, we do not consider MIX-INSTRUCT as our primary benchmark due to several limitations:

- **Limited Evaluation Metrics:** MIX-INSTRUCT uses BARTScore to measure the similarity between a model's output and a reference response generated by GPT-4. This approach conflates model quality with similarity to GPT-4, making it less suitable for evaluating true routing performance.



(a) Llama-2 7B and Llama-2 70B

(b) Mistral 7B and Llama-2 70B

Figure 12: Binary routing evaluation on Remastered Benchmark shows performance trade-offs across different model pairs.

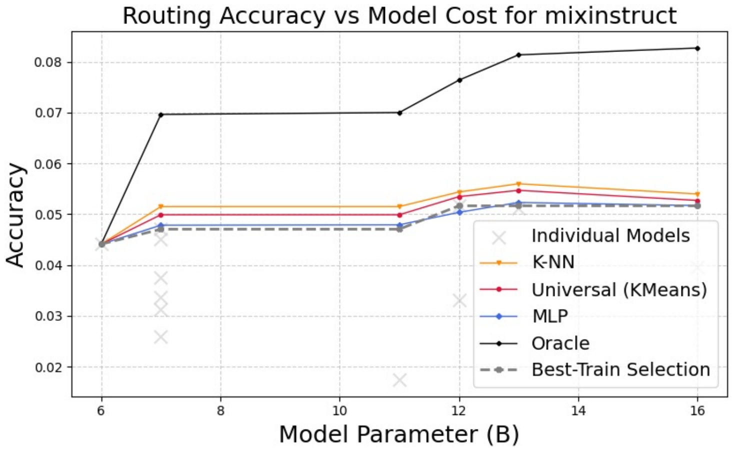


Figure 13: Routing performance on MIX-INSTRUCT.

It favors models that mimic GPT-4's phrasing—even when other models might generate more informative or appropriate responses—thus undermining the purpose of routing for capability-based model selection.

- **Limited Task Diversity:** The benchmark contains only five tasks, all of which fall under casual or instruction-following dialog. These tasks do not capture the breadth of real-world user queries, particularly in domains requiring specialized knowledge (e.g., science, math, law), thereby limiting the opportunity for routing to leverage model specialization.
- **Restricted Model Pool:** MIX-INSTRUCT covers about 10 models—comparable to RouterBench—restricting the expressiveness of routing policies. In contrast, EMBEDLLM benchmark includes over 100 models with diverse strengths while having some issues we listed in Section 2, offered a more realistic and rigorous setting for evaluating routing capabilities.