RouterArena: An Open Platform for Comprehensive Comparison of LLM Routers

Anonymous authors
Paper under double-blind review

Abstract

Today's LLM ecosystem comprises a wide spectrum of models that differ in size, capability, and cost. No single model is optimal for all scenarios; hence, LLM routers have become essential for selecting the most appropriate model under varying circumstances. However, the rapid emergence of various routers makes choosing the right one increasingly challenging. To address this problem, we need comprehensive router comparison and a standardized leaderboard, similar to those available for models. In this work, we introduce RouterArena, the first open platform enabling *comprehensive* comparison of LLM routers. RouterArena has (1) a principally constructed dataset with broad knowledge domain coverage, (2) distinguishable difficulty levels for each domain, (3) an extensive list of evaluation metrics, and (4) an automated framework for leaderboard updates. Leveraging our framework, we have produced the initial leaderboard with detailed metrics comparison as shown in Figure 1. We will make our platform open to the public; the current code base is available here: https://anonymous.4open.science/r/RouterArena.

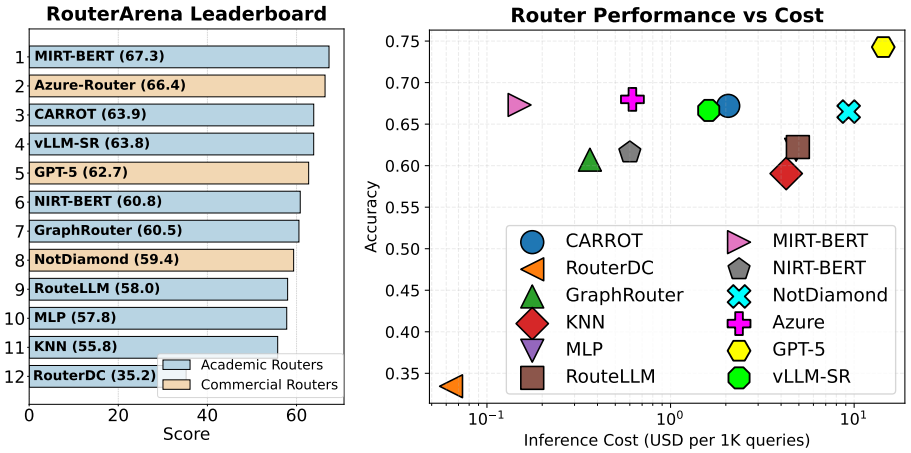


Figure 1: A quick view of RouterArena leaderboard and performance-cost trade-off.

1 Introduction

Large Language Models (LLMs) are rapidly diversifying, offering an ever-wider spectrum of capabilities and inference costs. This diversity increasingly challenges the prevailing LLM usage pattern in which users manually choose models for their queries. The difficulty stems from the fact that no single model is universally optimal: powerful models excel at complex tasks but are costly, while smaller models are more efficient yet may struggle on difficult queries. As a result, *LLM routers* that automatically select models based on input queries are increasingly recognized as a core system primitive in practical deployments.

Given its importance and promise, many LLM routers have recently emerged in both industry and academia (Figure 2). A notable example is GPT-5 (OpenAI, 2025), which incorporates routing as a key feature by directing user queries to the most suitable model within the OpenAI family. As

routers proliferate, the challenge shifts from selecting the right model to selecting the right router. Unfortunately, **router evaluation has not kept pace: there is currently no open evaluation platform**, akin to LMArena (Chiang et al., 2024), that systematically compares open-source routers (Hu et al., 2024; Zhuang et al., 2024a) and commercial routing services (NotDiamond, 2025; Microsoft, 2025) under a unified protocol.

It is urgent to fill this gap by building a *Router Arena* that can comprehensively evaluate and rank routers, enabling users to understand the status quo and make informed choices. However, unlike model arenas, designing a router arena is considerably more challenging due to the requirements from three key aspects. (1) **Dataset**. To evaluate whether a router can recognize problem domains and dispatch queries to appropriate models at minimal cost, the arena dataset must cover a broad range of domains and subjects, as well as varying difficulty levels. (2) **Metrics**. Router performance is inherently multi-dimensional, and so should be the arena ranking. While accuracy and cost are the primary metrics, it is also important to capture other dimensions such as routing optimality and robustness. (3) **Framework**. To enable live leaderboard updates, the arena must have a user-friendly framework that can automatically evaluate new open-source and commercial routers. Although existing studies explore some of these directions, as summarized in Table 1 and discussed in Section 2, they fail to address each challenge in a *comprehensive* way.

In this work, we present ROUTERARENA, the first open platform for comprehensive evaluation and comparison of LLM routers. It addresses the above key challenges with the following designs:

- **A Principled Diverse Dataset.** To ensure broad coverage, we construct the dataset using the Dewey Decimal Classification system adopted in libraries, covering all domains except religion. For each subject, we apply Bloom's taxonomy to design queries at three difficulty levels, producing a diverse dataset of $\sim$8,000 queries spanning 9 domains and 44 categories for router evaluation.
- **Extensive Metrics for Arena Ranking.** We construct router leaderboards by considering an extensive list of deployment-relevant metrics including query-answer accuracy, query-answer cost, routing optimality (cheapest correct selection), robustness to query perturbations (consistency), and router overhead (latency). This enables router comparison from multiple perspectives.
- **An Automated Framework for Leaderboard Updates.** We design a framework that automatically evaluates new routers, collects metrics, and updates the leaderboard. The framework supports both open-source and commercial routers, and employs prefix caching to improve efficiency.

Figure 1 provides a quick view of our accuracy–cost leaderboard along with other details. We have found that although GPT-5 achieves higher accuracy, its cost is significantly higher than that of other routers due to its model pool being restricted to the OpenAI family. Consequently, it does not rank as the best router on our accuracy–cost leaderboard.

Our vision is for ROUTERARENA to serve as an open community venue for evaluating routers as the ecosystem evolves, providing a standardized basis for fair comparison and progress tracking. By lowering the barrier to evaluation and enabling transparent, reproducible results, ROUTERARENA will help researchers and practitioners design, improve, and adopt better routers.

2 MOTIVATION

The Rapid Emergence of LLM Routers. As shown in Figure 2, the landscape of LLM routers is rapidly expanding, evolving from academic exploration to commercial deployment. From a few scattered academia routers (Zhang et al., 2023; Chen et al., 2023; Hari and Thomson, 2023) in mid-2023, the number of publications expanded to more than a dozen by 2024 (Liu et al., 2024; Zhuang et al., 2024b; Chen et al., 2024a; Zhao et al., 2024). By 2025, not only did academia routers continue to grow (Wang et al., 2025; Huang et al., 2025a; Ding et al., 2025; Zhang et al., 2025b), but commercial products also emerged (NotDiamond, 2025; Microsoft, 2025), most notably GPT-5 (OpenAI, 2025) with a built-in router.

New Problem: How to Choose the Right Router? Answering this question requires comprehensive router comparisons to understand the current landscape. Such comprehensiveness entails dataset categories, difficulty levels, evaluation metrics, and router inclusion. However, our review of existing work reveals that **no such comprehensive comparisons are available today**. As shown in Table 1, the existing work falls short in the following aspects.

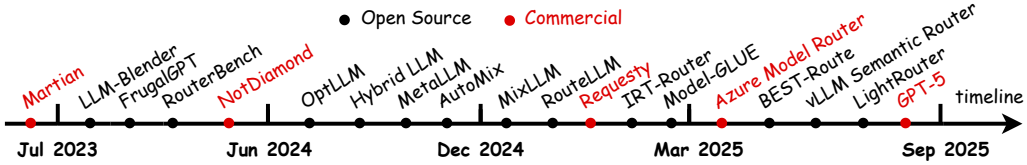


Figure 2: Timeline of example router-related works and products.

Table 1: Comparison of existing works (Hu et al., 2024; Huang et al., 2025b; Feng et al., 2025b; Zhuang et al., 2024a) and ROUTERARENA. ROUTERARENA enables comprehensive router comparison with extensive query categories, difficulty levels, evaluation metrics, and router inclusion.

Benchmark	Query categories	Difficulty levels	Evaluation Metrics	Commercial Routers	Router Ranking
RouterBench	24 Categories	✗ No analysis	✗ Deferral curve only	✗	✗
RouterEval	27 Categories	✗ No analysis	✗ Accuracy metric only	✗	✗
FusionBench	26 Categories	LLM-judge analysis	✗ Deferral curve only	✗	✗
EmbedLLM	26 Categories	✗ No analysis	✗ Accuracy metric only	✗	✗
ROUTERARENA	44 Categories based on DDC	✓ 5 Bloom Level Classification	✓ 5 Evaluation perspectives	✓ 3 Included	✓ Multi-metric leaderboard

- *Narrow Query Category Coverage.* They lack full coverage of query categories, making them impossible to evaluate router performance on queries from excluded categories.
- *Indistinguishable Difficulty Levels.* They do not differentiate queries by difficulty, limiting their ability to test accuracy–cost tradeoffs.
- *Incomplete Evaluation Metrics.* They only consider a subset of relevant metrics, overlooking important dimensions such as optimality, robustness, and latency.
- *No Support of Commercial Routers.* Current frameworks evaluate only open-source routers and do not extend to closed-source or commercial routers.
- *No Router Leaderboard.* There is no leaderboard that allows people to compare all routers under a unified evaluation protocol.

This Work: ROUTERARENA. This gap motivates us to design ROUTERARENA, an open platform for comprehensive router comparisons. In the remainder of this paper, we first introduce the key components of ROUTERARENA: principled dataset construction, comprehensive metric formulation, and an automated evaluation framework with live leaderboard. We then present our evaluation results and discuss the key findings.

3 ROUTERARENA EVALUATION DATASET

To enable meaningful and unbiased router comparisons, a high-quality evaluation dataset is essential. In this work, we construct such a dataset by adhering to two guiding principles for data collection.

Principle 1: DDC-Inspired Diverse Domain Coverage. To evaluate a router's ability to recognize problem domains and route queries to the appropriate specialist models, the dataset must provide broad domain coverage. To achieve this, we draw inspiration from the Dewey Decimal Classification (DDC) system (Dewey, 1876), a book classification framework widely used in libraries. The DDC is renowned for its comprehensive and logical structure, providing a proven methodology for organizing the entire world of knowledge into distinct, hierarchical categories (Satija, 2013).

Principle 2: Bloom-Based Distinguishable Difficulty Levels. To evaluate whether a router can determine query difficulty and make accuracy-cost tradeoffs—choosing between powerful but expensive models and weaker but cheaper ones—the dataset must include clearly distinguishable difficulty levels. To structure these difficulty levels, we adopt Bloom's taxonomy (Bloom et al., 1956), a widely used framework in quantifying question complexity (Ullrich and Geierhos, 2021; Herrmann-Werner et al., 2024; Padó, 2017). It works by classifying cognitive tasks into six ascending categories: remembering, understanding, applying, analyzing, evaluating, and creating. In this work, we further group these six levels into easy-medium-difficult three levels.

Dataset Construction Process. Following these two principles, we curate our evaluation dataset as follows. To ensure coverage across all DDC categories (excluding religion), we first collect all the queries from two existing LLM benchmark datasets and then supplement underrepresented categories with data from 21 open-source, domain-specific datasets. To determine the difficulty level of each query based on Bloom's taxonomy, we employ an LLM-as-Judge approach with DEEPSEEK-V3.1 (DeepSeek-AI, 2025) (prompt specified in Appendix D), enabling automatic difficulty annotation. We exclude the create-type questions because they are open-ended and cannot be reliably evaluated. We then categorize remembering and understanding questions as easy, applying questions as medium, and analyzing and evaluating questions as difficult.

Next, to fairly distribute questions across categories and difficulty levels, we propose a recursive deficit redistribution algorithm. We begin by setting the ratio of science to humanities at 2:1. Within each top-level category, if a sub-category falls short of its proportional quota, the resulting surplus is recursively and uniformly redistributed to those sub-categories that exceed their initial allocation. We apply the same procedure within each sub-category to allocate data across different difficulty levels, ensuring balanced coverage throughout the dataset.

The above steps yield approximately 62,000 queries in total. However, this raw dataset contains many highly similar or even duplicate questions inherited from the various sampled sources. Such redundancy does not benefit router evaluation and may even introduce noise into the results. To address this, we perform cosine-similarity–based de-duplication using SENTENCE-TRANSFORMERS/ALL-MINILM-L6-V2. By strictly following the allocation strategy and selecting the least similar samples, we ensure that the resulting ROUTERARENA dataset maintains broad coverage with minimal redundancy.

The Resulting Dataset. Our final evaluation dataset consists of 8,400 queries sampled from 23 source datasets. It spans nine top-level domains and 44 categories, with each category containing queries across three difficulty levels. Figure 3 illustrates the detailed composition of the dataset. Note that the distribution of difficulty levels is skewed, but it reflects real-world query patterns—easy and medium questions occurring more frequently than hard ones. We include more details about the dataset in Appendix C, including dataset schema and concrete query examples.

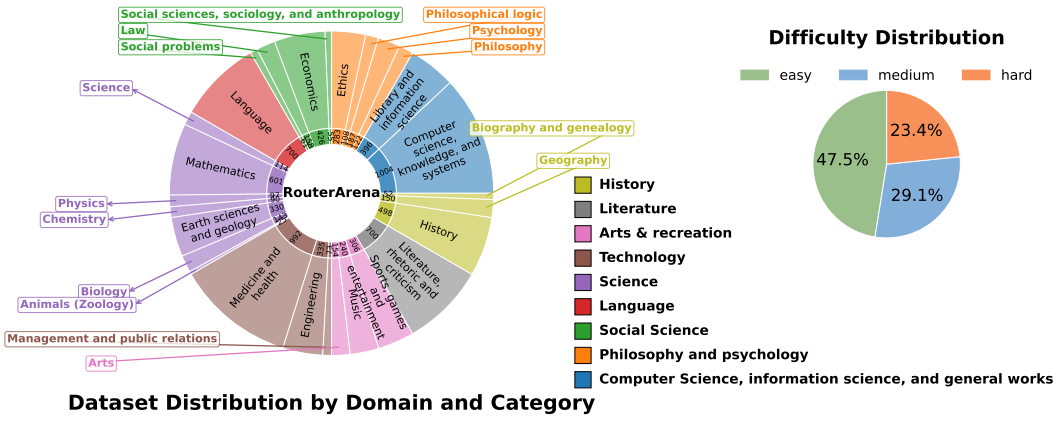


Figure 3: Dataset composition. For ease of demonstration, we merged some categories.

4 ROUTERARENA EVALUATION METRICS

ROUTERARENA supports comprehensive router evaluation along five dimensions.

(1) Query-answer Accuracy. This metric captures a router's ability to direct queries to the appropriate models such that they are correctly answered. We calculate accuracy as the average correctness across all our dataset queries.

(2) Query-answer Cost. This measures the cost incurred by a router's routing decisions. To address important factors such as the variable cost introduced by input length and generation length

(e.g., in chain-of-thought reasoning) as well as the distinct computational characteristics of Mixture-of-Experts (MoE) models, we use the actual inference cost measured by:

$$cost = c_{in} * N_{in} + c_{out} * N_{out}$$

Where c is the cost per token and N is the number of tokens. We obtain the cost c for the specific models a router chooses using the official API pricing published by the corresponding providers (e.g., OpenAI, Claude, Fireworks AI, etc.). For unpopular models that are not served by commercial providers, we deploy them ourselves for experiments (only a few). In such cases, we approximate their costs using the pricing tiers published by commercial hosting platforms (e.g., Together.ai), which estimate serving costs based on model size (parameter count) and architecture type (e.g., MoE). Table 5 summarizes the pricing tiers we use for our self-hosted model.

(3) Routing Optimality. This captures a router's ability to perform optimal routing—that is, selecting the cheapest model that still produces a correct response. It consists of three sub-metrics: *(a) Optimal Selection Ratio*—the proportion of queries for which the router answers correctly by selecting the cheapest model; *(b) Optimal Accuracy Ratio*—the ratio between a router's achieved accuracy and the upper-bound accuracy obtainable when always selecting the best model from its model pool; *(c) Optimal Cost Ratio*—the ratio between the cost incurred by the router's selections and the cost of always choosing the optimal model. This metric will penalize routers that rely on unnecessarily expensive models when cheaper, correct alternatives are available. This metric will penalize routers that use unnecessarily expensive models when cheaper, correct alternatives are available.

(4) Routing Robustness. This metric evaluates the router's robustness against noisy inputs. We calculate it as the proportion of queries for which the router makes consistent routing decisions under perturbed input. Specifically, we generate noisy variants of queries—through paraphrasing, grammatical changes, synonym substitutions, and typos—and measure the percentage of cases where the router selects the same model as it does for the original, noise-free query. This captures the router's capability for handling realistic, imperfect user queries.

(5) Routing Latency. Since the router operates in the critical path of systems in production, it must handle millions of queries per second with minimal overhead. This metric measures the additional latency introduced by routing. It reflects the latency increase in both time-to-first-token (TTFT) and end-to-end response latency when a given router is employed.

5 RouterArena Evaluation Framework

5.1 Arena Ranking

RouterArena provides a series of router leaderboards that enables users to compare the capabilities of different routers and select the one best suited to their scenarios. It includes six ranking scores based on the evaluation metrics described in Section 4, including Arena, Optimal-selection-ratio, Optimal-acc-ratio, Optimal-cost-ratio, Robustness, and Latency. Among these, the Arena score captures the trade-off between accuracy and cost by combining them into a single composite measure using the Weighted Harmonic Mean (Ferger, 1931). Specifically, to better distinguish between routers with low costs, we apply a base-2 logarithmic (log_2) transformation to the cost values. Under this scaling, each doubling of price reduces the cost score by one unit. For router i with cost c_i, we define its normalized cost as

$$C_i = \frac{log_2(c_{max}) - log_2 c_i}{log_2(c_{max}) - log_2(c_{min})}$$

where $c_{\max}$ and $c_{\min}$ denote the maximum and minimum costs of routing 1k queries. Specifically, we choose $c_{min} = 0.0044$, corresponding to the cost of the cheapest model in the leaderboard's model pool. This reflects the cost of a router that always selects the cheapest model. We choose $c_{max} = 200$ representing the most expansive model, OpenAI's o1-pro. This normalization maps the cost into range $[0, 1]$, with larger values of C_i corresponding to more economical routers. Next, we combine the normalized cost C_i and accuracy A_i using a weighted harmonic mean:

$$S_{i,\beta} = \frac{(1+\beta)A_i C_i}{\beta A_i + C_i},$$

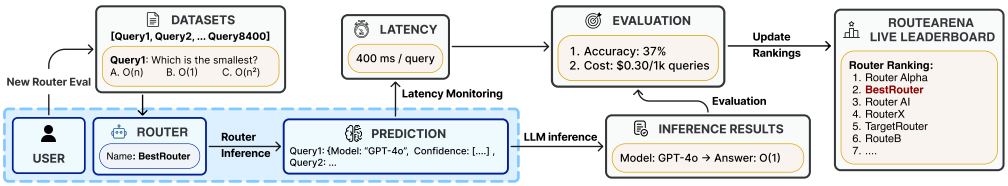


Figure 4: RouterArena Live Leaderboard.

where the parameter $\beta > 0$ controls the relative importance of accuracy versus cost. Setting $\beta > 1$ places greater weight on cost. By default, we use $\beta = 0.1$, emphasizing routing accuracy, because highly accurate routers are generally more valuable even if they incur slightly higher costs.

5.2 Automated Evaluation Framework

Although we demonstrate RouterArena with a specific set of routers in this paper, it is very easy to update the leaderboard with new routers. To facilitate this process, we have designed an automated evaluation framework that will be released publicly alongside the leaderboard. Figure 4 shows the overall system workflow. To evaluate a new router, the user can simply start by providing our framework with an access point (e.g., an API) to the router. The framework sends evaluation queries to the router, which performs routing inference on its end and returns its model selections. To ensure fairness, we run the inference ourselves and use cached results when possible, since many routers share overlapping model pools. During this process, the router's response time is monitored to measure routing latency. Finally, the framework computes the evaluation metrics and aggregates the results, which are reflected in the leaderboard. Note that some commercial routers may not expose their model selections and instead return query answers directly. Such routers can still be evaluated with our framework, although certain metrics cannot be measured in this setting.

6 Experiments

6.1 Experimental Settings

Router Selection For commercial routers, we evaluated the router from Not Diamond (NotDiamond, 2025), which provides access to over 60 models, and the Azure Model Router (Microsoft, 2025), which currently only supports OpenAI models. We also included GPT-5 (OpenAI, 2025), whose model family incorporates an internal router. For NotDiamond, we selected 26 representative models spanning different parameter scales, architectures, and reasoning abilities. For AzureRouter, we evaluated the entire model pool, including GPT-5 model families. Appendix B provides full details of the model pools used for each router.

For open-source routers, we evaluated nine representative systems covering a diverse routing approaches. Specifically, we chose both the KNN- and MLP-based methods trained on RouterBench (Hu et al., 2024) as baselines. We further included GraphRouter (Feng et al., 2025a), which leverages graph neural networks (GNNs) for routing, and the Universal Router (Jitkrittum et al., 2025), which uses K-means clustering. To capture cost–accuracy tradeoffs, we evaluated CARROT Router (Somerstep et al., 2025), while RouterDC (Chen et al., 2024b) was incorporated as a dual contrastive learning–based approach. Additionally, we considered IRT-Router (Song et al., 2025), which applies item response theory to explicitly model the interaction between query attributes and model capabilities, and RouteLLM (Ong et al., 2025), which performs binary selection between a stronger and a weaker model. Moreover, we also take the latest vLLM Semantic Router (vLLM, 2025) into consideration, which leverages a ModernBERT (Hugging Face, 2025) to categorize the incoming requests into pre-defined categories, and selects the model that has the highest score.

Router Training and Evaluation For commercial routers, no additional training is required; we simply accessed their provided APIs for evaluation. In contrast, for academia routers, we followed the training procedures and datasets specified in their open-source implementations. Specifically, we did not modify the training datasets or the task categorizations (if applicable). The model pools were configured in accordance with the original papers. In particular, for the vLLM-SR, we constructed the pool using both open-source models of varying parameter scales and proprietary models, with

detailed configurations summarized in Table 3. After training each router, we evaluated them by feeding our benchmark dataset, recording the model selected, the latency incurred by the selection, and the confidence scores assigned to all candidate models in the pool.

6.2 RESULTS

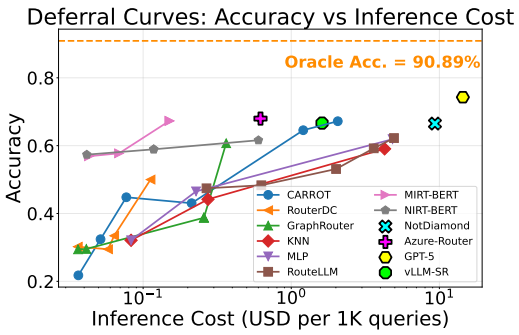


Figure 5: Deferral Curve: accuracy versus cost

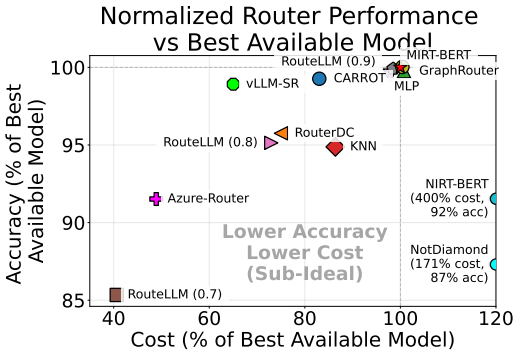


Figure 6: Normalized Deferral Curve

Deferral Curve. Figure 5 presents the trade-off between accuracy and inference cost. As we increase inference budget, we unlock more powerful models, driving the accuracy up. For open-source routers, we leveraged their confidence scores to apply budget-based masking, which produces multiple points along each curve. With only cheap models available, accuracy remains low, but as larger models enter the pool, routing accuracy increases. In contrast, commercial routers typically appear as single points because their model pools already include the best-performing models.

Two insights emerge. First, the orange dashed line shows the oracle accuracy, highlighting that all routers fall short of the best achievable performance. Second, the trade-off frontier differs by setting: commercial routers can achieve higher accuracy, but usually at significantly higher costs; open-source routers, on the other hand, achieve competitive performance at much lower budgets, though they plateau earlier. Notably, routers like CARROT and GraphRouter illustrate cost-efficient routing, while systems such as GPT-5 and NotDiamond lean heavily on expensive models for accuracy. This suggests that while commercial routers prioritize maximizing accuracy, academic approaches often explore the efficiency side of the frontier.

Normalized Deferral Curve. Figure 6 reports router accuracy and cost normalized to each router's best-performing model, point (100%, 100%) on the plot. The upper-left quadrant represents the ideal case—higher accuracy with lower cost by leveraging smaller models. In practice, most routers cluster near the baseline (100% cost, 100% accuracy), suggesting they over-rely on the strongest model and miss opportunities to defer to cheaper alternatives. Notably, NIRT-BERT illustrates inefficiency, reaching only baseline-level accuracy while incurring 378% of the cost.

By contrast, routers such as vLLM-SR and CARROT achieve meaningful savings: roughly 35% lower cost with under 2% accuracy degradation. These cases show routing can indeed improve efficiency when smaller models are effectively utilized. Overall, the figure highlights a clear trade-off—higher accuracy often comes with higher cost—while also pointing to promising directions for designing routers that move closer to the ideal frontier.

Optimality Score. Figure 7 highlights the inherent trade-off between routing accuracy and cost. In practice, routers that achieve higher accuracy typically do so at the expense of a higher cost ratio, since they defer more often to large, expensive models. This behavior lowers their optimal selection ratio, i.e., the frequency with which they choose the most efficient model for each query. This pattern is most apparent in the binary routers such as RouteLLM. By design, these routers face a sharp trade-off: they achieve higher accuracy by routing more queries to the stronger model, which drives up cost. In contrast, multi-model routers have a more flexible pool, and while the general trend still holds, we see greater variability depending on pool composition and routing strategy. Among non-binary routers, RouterDC stands out with the lowest cost ratio and highest optimal selection ratio, but this comes at the cost of poor overall accuracy. At the other extreme, MIRT-BERT achieves strong

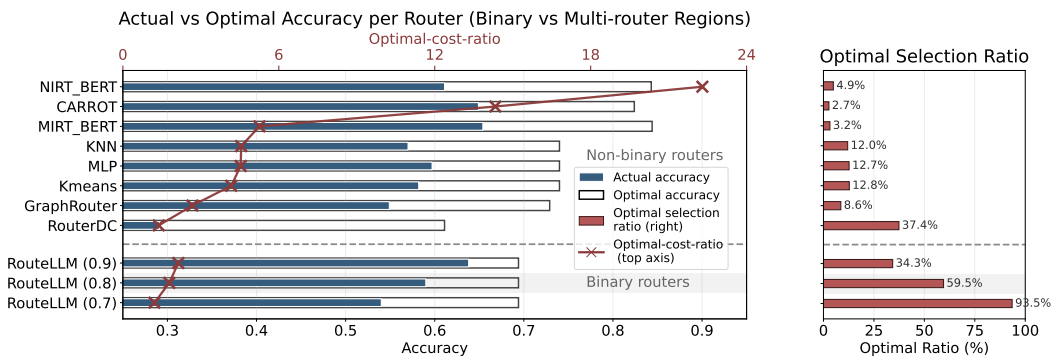


Figure 7: Actual and optimal accuracy, along with optimal selection ratio and cost ratio

accuracy (close to 77% of its optimal accuracy) but requires nearly five times the optimal cost, placing it closer to the "high-cost high-accuracy" region of the trade-off frontier. In other words, while some routers are closer to the efficiency frontier than others, none simultaneously combine low cost and high accuracy. Overall, our findings indicate that current routing methods have learned to leverage large models to boost performance, but remain inefficient at recognizing when smaller models are sufficient. This creates a clear opportunity for future work: developing routers that can balance accuracy and efficiency by selectively deferring to large models only when necessary.

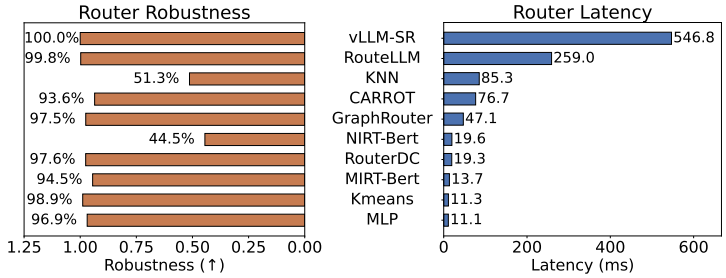


Figure 8: Router robustness and latency comparison.

Robustness and Latency. Given that user prompts are often noisy, we further assess router sensitivity and robustness. Specifically, we prepend an irrelevant keyword to the input and observe whether the router alters its original model selection. We define robustness as 1 – the proportion of changed selections. As shown in Figure 8, routers leveraging latent representations of prompts demonstrate stronger stability against noise, whereas methods relying on explicit representations, such as KNN and NIRT-BERT, are considerably more sensitive. These findings highlight the importance of applying prompt engineering techniques to mitigate noisy queries in the future.

Furthermore, Figure 8 reports the end-to-end latency of routers on a single A100 GPU, measured from the time a request is received to the output of the model selection result. Among all methods, vLLM-SR and RouteLLM exhibit significantly higher latency because they rely on the OpenAI embedding API, which introduces additional network delays. In contrast, other routers consistently maintain sub-100ms latencies. Since the LLM router lies on the critical path of the end-to-end systems, our results provide new insights for industrial deployment: while LLM routers can optimize accuracy and cost, they also introduce non-negligible overhead that may even compromise service-level objectives (SLOs).

6.3 Insights from the Final RouterArena Leaderboard.

Our evaluation produces the router leaderboard shown in Table 2. The leaderboard consists of six ranking scores, and the overall ranking is determined by averaging across them. We highlight two key findings: (1) Commercial routers do not necessarily outperform open-source routers. For example, GPT-5 ranks #7 due to its restricted model pool, and NotDiamond ranks #12 because it frequently selects expensive models. (2) No router ranks at the top across all metrics, reflecting the inherent trade-offs in router design.

Table 2: Ranking of routers across multiple metrics. Lower values indicate better performance.

#	Router	Arena Rank	Optimal-selection-ratio Rank	Optimal-cost-ratio Rank	Optimal-acc-ratio Rank	Robustness Rank	Latency Rank	Average
1	Azure-Router	2	-	-	-	-	-	2
2	RouteLLM	9	2	2	1	2	8	4
3	MLP	10	3	4	2	5	1	4.17
4	MIRT-BERT	1	7	6	4	6	2	4.33
5	vLLM-SR	4	-	-	-	1	9	4.67
6	RouterDC	12	1	1	8	3	**3**	4.67
7	GPT-5	5	-	-	-	-	-	5
8	GraphRouter	7	6	3	6	4	5	5.17
9	CARROT	3	8	8	3	7	6	5.83
10	NIRT-BERT	6	5	7	7	9	4	6.33
11	KNN	11	4	5	5	8	7	6.67
12	NotDiamond	8	-	-	-	-	-	8

For developers and researchers, the findings highlight key deficiencies in current routing methods and point toward clear directions for designing the next generation of routers. The results show that all existing routers fall short of the oracle's achievable performance, primarily because they are inefficient at recognizing when smaller, cheaper models are sufficient for a given query. Future work should focus on closing this performance gap. Moreover, the high latency and poor robustness of certain routers open new avenues of research beyond the traditional cost-versus-accuracy trade-off. Developers can use the platform's automated framework to submit and benchmark new routers against established leaders, fostering innovation and transparently tracking progress in the field.

7 Related Work

LLM Router. With the increasing availability of specialized models that can surpass even the most capable general-purpose LLMs in specific domains, both academia and industry have been actively exploring how to build LLM routers. In industry, several systems have emerged. Martian Router (WithMartian, 2025) proposed the idea of model mapping, while Storytell (Storytell.ai, 2025) categorizes user queries and routes them to the best-performing models. Other companies also seek to find the optimal model for user's tasks by balancing performance and cost (NotDiamond, 2025; RequestyAI, 2025; OpenAI, 2025). Recent academic efforts have also begun to emerge. GraphRouter (Feng et al., 2025a) leverages graph neural networks, and Router-R1 (Zhang et al., 2025a) employs reinforcement learning. The growth of open-source solutions underscores the need for effective router evaluation (Somerstep et al., 2025; Song et al., 2025; Feng et al., 2025b).

LLM Router Benchmark. RouterBench (Hu et al., 2024) introduces a large-scale dataset consisting of over 405k inference outcomes from representative LLMs. RouterEval (Huang et al., 2025b) collects performance results from 8,500 LLMs across 12 widely used benchmarks. FusionBench (Feng et al., 2025b) covers 14 tasks across five domains and leverages 20 open-source LLMs. Other benchmarks have also contributed to this line of work by using different data collection methods (Kassem et al., 2025; Mei et al., 2025). However, these benchmarks fail to provide broad coverage across disciplinary domains or cover all kinds of routers. In contrast, our benchmark is systematically constructed based on an authoritative knowledge classification framework, making it the first comprehensive and actionable benchmark for LLM routing.

8 Conclusion

We introduce RouterArena, the first open platform for comprehensive router comparison. Our platform features a principled dataset with broad domain coverage and varying difficulty levels, an extensive set of evaluation metrics, and an automated framework to maintain a live leaderboard. Initial evaluations of 12 routers reveal a significant trade-off between accuracy and cost, showing that no single router is universally optimal. Commercial routers tend to achieve higher accuracy at a much greater expense, while open-source routers often present more cost-efficient solutions. Overall, our findings indicate that current routers are inefficient at leveraging cheaper models when appropriate, highlighting a clear opportunity for future work.

ETHICS STATEMENT

All authors have read and will adhere to the ICLR Code of Ethics (https://iclr.cc/public/CodeOfEthics). Our evaluation dataset was constructed by aggregating and sampling from publicly available and open-source datasets. To ensure broad topic coverage while avoiding potentially sensitive subjects, we based our domain selection on the Dewey Decimal Classification system, explicitly excluding the category of religion. The difficulty level of each query was annotated using an "LLM-as-Judge" approach. We acknowledge that this process, along with the use of existing datasets, may introduce or perpetuate biases inherent in the source data and the annotator model. We have made the dataset and our collection methodology public to allow for further inspection and bias analysis by the community.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the full reproducibility of our work. All components of the ROUTERARENA platform, including the dataset, evaluation framework, and code, will be made publicly available.

(1) Dataset: The principles and detailed process for our dataset construction are described in Section 3. This includes domain coverage strategy, difficulty level annotation, and our deduplication process. Further details on the dataset schema, composition, and examples are provided in Appendix C. (2) Evaluation: Our five evaluation metrics are precisely defined in Section 4, and the formula for the composite leaderboard score is detailed in Section 5.1. The automated evaluation framework is described in Section 5.2. (3) Experiments: The specific academic and commercial routers evaluated are listed in Section 6.1.1. The exact model pools used for each router are provided in Appendix B. For all academic routers, we followed the training procedures and configurations specified in their original open-source implementations.

The public release of our complete framework will enable researchers to replicate our results and evaluate new routers on the leaderboard.

REFERENCES

Benjamin S Bloom et al. Taxonomy of. *Educational Objectives*, 250, 1956.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328, 2024a.

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, 2024b. URL https://arxiv.org/abs/2409.19886.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.

Yu-Neng Chuang, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, Xia Hu, and Helen Zhou. Learning to route llms with confidence tokens. *arXiv preprint arXiv:2410.13284*, 2024.

DeepSeek-AI. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Melvil Dewey. *A classification and subject index, for cataloguing and arranging the books and pamphlets of a library*. Brick row book shop, Incorporated, 1876.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks VS Lakshmanan, Qingyun Wu, and Victor Rühle. Best-route: Adaptive llm routing with test-time optimal compute. *arXiv preprint arXiv:2506.22716*, 2025.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections, 2025a. URL https://arxiv.org/abs/2410.03834.

Tao Feng, Haozhen Zhang, Zijie Lei, Pengrui Han, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Jiaxuan You. Fusing llm capabilities with routing data, 2025b. URL https://arxiv.org/abs/2507.10540.

Wirth F Ferger. The nature and use of the harmonic mean. *Journal of the American Statistical Association*, 26(173):36–40, 1931.

Surya Narayanan Hari and Matt Thomson. Tryage: Real-time, intelligent routing of user prompts to large language models. *arXiv preprint arXiv:2308.11601*, 2023.

Annette Herrmann-Werner, Tanja Festl-Wietek, Friederike Holderried, Lena Herschbach, Jan Griewatz, Ken Masters, Stephan Zipfel, and Matthias Mahling. Assessing ChatGPT's mastery of Bloom's Taxonomy using psychosomatic medicine exam questions: Mixed-methods study. *Journal of Medical Internet Research*, 26:e52113, 2024. doi: 10.2196/52113. URL https://doi.org/10.2196/52113.

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. URL https://arxiv.org/abs/2403.12031.

Zhongzhan Huang, Guoming Ling, 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*, 2025a.

Zhongzhan Huang, Guoming Ling, 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, 2025b. URL https://arxiv.org/abs/2503.10657.

Hugging Face. Modernbert: A new era of pretraining. https://huggingface.co/blog/modernbert, 2025. Hugging Face Blog, accessed: Sep. 2025.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient llm inference, 2025. URL https://arxiv.org/abs/2502.08773.

Aly M. Kassem, Bernhard Schölkopf, and Zhijing Jin. How robust are router-llms? analysis of the fragility of llm routing capabilities, 2025. URL https://arxiv.org/abs/2504.07113.

Yueyue Liu, Hongyu Zhang, Yuantian Miao, Van-Hoang Le, and Zhiqiang Li. Optllm: Optimal assignment of queries to large language models. In *2024 IEEE International Conference on Web Services (ICWS)*, pages 788–798. IEEE, 2024.

Kai Mei, Wujiang Xu, Shuhang Lin, and Yongfeng Zhang. Omnirouter: Budget and performance controllable multi-llm routing, 2025. URL https://arxiv.org/abs/2502.20576.

Microsoft. Model router for azure ai foundry (preview) concepts. https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/model-router, 2025. Accessed: Sep 2025.

NotDiamond. Notdiamond: Routing between ai models. https://www.notdiamond.ai/about, 2025. Accessed: Sep. 2025.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025. URL https://arxiv.org/abs/2406.18665.

OpenAI. Gpt-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf, August 2025. Accessed: Sep 2025¿.

Ulrike Padó. Question difficulty – how to estimate without norming, how to use for automated grading. In Joel Tetreault, Jill Burstein, Claudia Leacock, and Helen Yannakoudakis, editors, *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–10, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5001. URL https://aclanthology.org/W17-5001/.

RequestyAI. Requestyai: Unified llm gateway routing. https://www.requesty.ai/solution/llm-routing, 2025. Accessed: Sep. 2025.

Mohinder Partap Satija. *The theory and practice of the Dewey decimal classification system*. Elsevier, 2013.

Seamus Somerstep, Felipe Maia Polo, Allysson Flavio Melo de Oliveira, Prattyush Mangal, Mírian Silva, Onkar Bhardwaj, Mikhail Yurochkin, and Subha Maity. Carrot: A cost aware rate optimal router, 2025. URL https://arxiv.org/abs/2502.03261.

Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. Irt-router: Effective and interpretable multi-llm routing via item response theory, 2025. URL https://arxiv.org/abs/2506.01048.

Storytell.ai. Storytell.ai: Dynamic llm routing. https://web.storytell.ai/llm-router, 2025. Accessed: Sep. 2025.

Sabine Ullrich and Michaela Geierhos. Using bloom's taxonomy to classify question complexity. In Mourad Abbas and Abed Alhakim Freihat, editors, *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 285–289, Trento, Italy, 12–13 November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.icnlsp-1.34/.

vLLM. vllm semantic router. https://vllm-semantic-router.com/, 2025. Accessed: Sep. 2025.

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.

WithMartian. Martian model router. https://docs.withmartian.com/martian-model-router, 2025. Accessed: Sep. 2025.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning, 2025a. URL https://arxiv.org/abs/2506.09033.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. Model spider: Learning to rank pre-trained models efficiently. *Advances in Neural Information Processing Systems*, 36:13692–13719, 2023.

Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Capability instruction tuning: A new paradigm for dynamic llm routing. *arXiv preprint arXiv:2502.17282*, 2025b.

Xinyu Zhao, Guoheng Sun, Ruisi Cai, Yukun Zhou, Pingzhi Li, Peihao Wang, Bowen Tan, Yexiao He, Li Chen, Yi Liang, et al. Model-glue: Democratized llm scaling for a large model zoo in the wild. *Advances in Neural Information Processing Systems*, 37:13349–13371, 2024.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. Embedllm: Learning compact representations of large language models, 2024a. URL https://arxiv.org/abs/2410.02223.

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*, 2024b.

A THE USE OF LLMS

In this work, Large Language Models (LLMs) were used in two distinct capacities: as a core component of our research methodology and as a general-purpose writing aid.

LLM-as-Judge for Data Annotation: A significant use of an LLM was in the construction of our evaluation dataset. As detailed in Section 3, we employed DeepSeek-V3.1 as an "LLM-as-Judge" to automatically annotate the difficulty of each query according to Bloom's taxonomy. This automated approach allowed us to systematically label a large and diverse set of questions. The specific prompt used for this annotation process is provided in Appendix D.

Writing Assistance: An LLM was also utilized as a general-purpose tool to assist in the writing process. Its role was limited to proofreading the manuscript for grammatical errors, improving clarity, and ensuring stylistic consistency. The LLM was not used for research ideation, conducting experiments, or writing the core scientific contributions of the paper.

B MODEL POOLS BY ROUTER

Table 3: Model pools used by different routers.

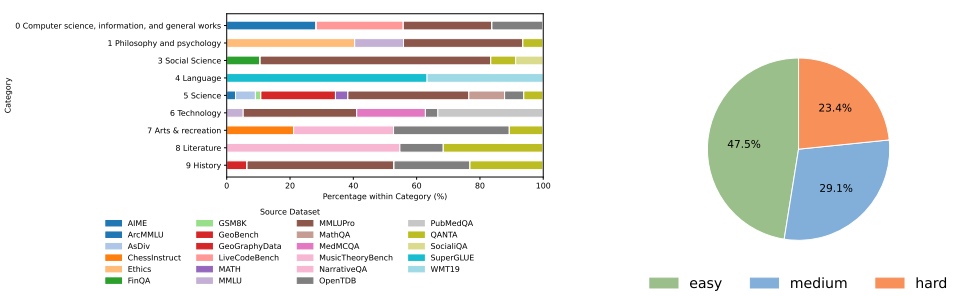
Router	**Model Pool**
RouterBench	WizardLM/WizardLM-13B-V1.2; claude-instant-v1; claude-v1; claude-v2; gpt-3.5-turbo-1106; gpt-4-1106-preview; meta/codellama-34b-instruct; meta/llama-2-70b-chat; mistralai/mistral-7b-chat; mistralai/mixtral-8x7b-chat; zero-one-ai/Yi-34B-Chat
GraphRouter	meta-llama/llama-3-8b-instruct; mistralai/mixtral-8x7b-chat; nousresearch/nous-34b-chat; meta/llama-2-7b-chat; mistralai/mistral-7b-chat; meta/llama-3-70b-chat; meta/llama-3-turbo-8b-chat; meta/llama-3-turbo-70b-chat; meta/llama-3.1-turbo-70b-chat; qwen/qwen-1.5-72b-chat
Universal	WizardLM/WizardLM-13B-V1.2; claude-instant-v1; claude-v1; claude-v2; gpt-3.5-turbo-1106; gpt-4-1106-preview; meta/codellama-34b-instruct; meta/llama-2-70b-chat; mistralai/mistral-7b-chat; mistralai/mixtral-8x7b-chat; zero-one-ai/Yi-34B-Chat
CarrotRouter	aws-claude-3-5-sonnet-v1; aws-titan-text-premier-v1; openai-gpt-4o; openai-gpt-4o-mini; wxai-granite-3-2b-instruct-8k-max-tokens; wxai-granite-3-8b-instruct-8k-max-tokens; wxai-llama-3-1-70b-instruct; wxai-llama-3-1-8b-instruct; wxai-llama-3-2-1b-instruct; wxai-llama-3-2-3b-instruct; wxai-llama-3-3-70b-instruct; wxai-mixtral-8x7b-instruct-v01; wxai-llama-3-405b-instruct
RouterDC	mistralai/Mistral-7B-v0.1; meta-math/MetaMath-Mistral-7B; itpossible/Chinese-Mistral-7B-v0.1; HuggingFaceH4/zephyr-7b-beta; cognitivecomputations/dolphin-2.6-mistral-7b; meta-llama/llama-3-8b-instruct; cognitivecomputations/dolphin-2.9-llama3-8b
IRT-Router	glm_4_air; glm_4_flash; glm_4_plus; gpt_4o; gpt_4o_mini; gpt_4o_mini_cot; deepseek_coder; deepseek_chat; qwen25_32b_int4; qwen25_7b_instruct; qwen25_72b_instruct; qwq_32b_preview; qwen25_math_7b_instruct; llama31_8b_instruct; llama31_70b_instruct; llama31_405b_instruct; mixtral_8x7b_instruct; mistral_7b_instruct_v02; ministral_8b_instruct_2410; gemini15_flash; claude35_haiku20241022
RouteLLM	openai-gpt-4o; mixtral_8x7b_instruct

We provide the model pool used by each router here.

C DATASETS DETAILS

Figure 9 illustrates the domain coverage of our dataset across the nine Dewey Decimal categories. Each horizontal bar represents the relative contribution of different source datasets within a category. This distribution highlights the systematic integration of multiple datasets to achieve a more balanced representation of both general-purpose and highly specialized domains.

Table 4 shows the detailed dataset columns.



(a) Dataset source.

(b) Difficulty distribution.

Figure 9: Overview of dataset composition: (a) data sources and (b) difficulty distribution.

Table 4: Overview of dataset columns

Column	Description	Example
Category	Bloom's taxonomy high-level class	9 History
Sub Category	Bloom's taxonomy sub-class	02 Library and information sciences
Dataset Name	Source dataset	ArcMMLU
Global Index	Unique instance ID	ArcMMLU_114
Context	Supporting passage (if any)	Sasha decided to watch TV and get some food
Question	Input question	What is the capital of France?
Options	Multiple-choice options	["10", "20", "30", "40"]
Answer	Ground-truth answer	Paris
Bloom Level	Bloom's taxonomy difficulty level	Understanding

C.1 Dataset Examples

Domain: 0 Computer science, information, and general works
Category: 02 Library and information sciences
Dataset name: ArcMMLU
Global Index: ArcMMLU_170
Context: ""
Question: "In Which five year plan the INFLIBNET' was established ( )."
Options: ["Fourth five year plan", "Fifth five year plan", "Sixth five year plan", "Seventh five year plan"]
Answer: D
Bloom_level: remeber

Domain: 4 Language
Category: 40 Language
Dataset name: SuperGLUE-CausalReasoning
Global Index: SuperGLUE-CausalReasoning_4513
Context: "Political violence broke out in the nation."
Question: "what's the effect of this?"
Options: ["Many citizens relocated to the capitol.", "Many citizens took refuge in other territories."]
Answer: 1.0
Bloom_level: analyze

Figure 10: Dataset examples.

Figure 10 shows some dataset examples.

D LLM Prompts

D.1 Bloom Taxonomy Prompt

We used the following prompt for classifying each question into different Bloom Levels:

Instruction: You are an evaluator tasked with classifying questions by cognitive difficulty using **Bloom's Taxonomy (Revised 2001)**. Bloom's Taxonomy defines six levels of cognitive processes:

1. **Remember** – Recall or recognize facts, terms, or concepts. 2. **Understand** – Explain, summarize, interpret, or demonstrate comprehension. 3. **Apply** – Use learned knowledge to solve problems in new or routine situations. 4. **Analyze** – Break down information, examine parts, relationships, or underlying causes. 5. **Evaluate** – Make judgments or decisions based on evidence, criteria, or standards. 6. **Create** – Put elements together to form a new structure, idea, or product.

Your Task: Given the details of a question, determine *which Bloom's level best represents the cognitive process required to answer it*.

- Output the result in a **structured JSON block** with no additional text. - If multiple levels might apply, choose the *highest* level required.

Output Format:

```
{
  "bloom_level": "<Remember | Understand | Apply | Analyze |
  Evaluate | Create>"
}
```

Input Fields:

```
- Dataset name: {dataset_name}
- Dataset classification (DDC category): {dataset_category}
- Question: {question}
- Question Index: {question_index}
- Context (if any): {context}
- Options (if MCQ): {options}
- Answer: {answer}
```

D.2 Evaluation Prompt

The following is an example of an evaluation prompt we use:

```
Please read the following multiple-choice questions and
consider which option best follows deontological ethics.\n\n
Context: {Context}\n\n
Question: {Question}\n\n
Options: \n{Options}\n\n
Provide the correct letter choice in \\boxed{{X}}, where X is
the correct letter choice.
Keep the explanation or feedback within 3 sentences.
```

E Leaderboard

Figure 11 presents the spider plot of RouterArena, which compares six routing methods (CARROT, RouterDC, GraphRouter, MIRT-BERT, NIRT-BERT, and RouteLLM) across five evaluation dimensions: Arena Score, Cost-ratio Score, Optimal-acc Score, Latency Score, and Robustness Score. Each axis indicates higher performance in the outward direction, allowing a direct visualization of trade-offs. For example, CARROT achieves strong performance in Arena and Latency Scores, while RouterDC excels in Cost-ratio Score.

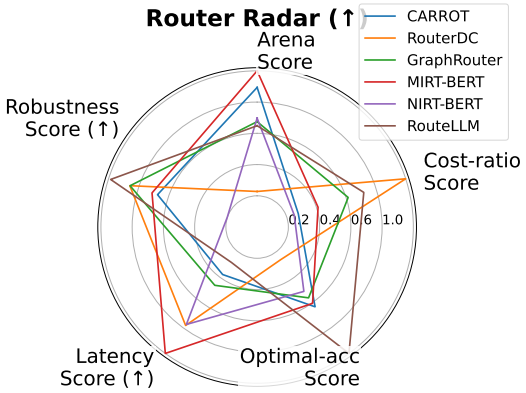


Figure 11: The spider plot of ROUTERARENA

Table 5: Dense and MoE large language model size and price per million tokens.

Dense Models		**Mixture-of-Experts Models**	
Model Size	**Price**	**Model Size**	**Price**
Up to 4B	$0.10	Up to 56B	$0.60
4.1B – 8B	$0.20	56.1B – 176B	$1.20
8.1B – 21B	$0.30	176.1B – 480B	$2.40
21.1B – 41B	$0.80		
41.1B – 80B	$0.90		
80.1B – 110B	$1.80		