RouteJudge: Preference-Based Evaluation of LLM Routers under Pluralistic User Preferences

Guannan Lai [1 2] **Haoran Hu** [1 2] **Han-Jia Ye** [1 2]

Abstract

Existing evaluations of large language model (LLM) routing systems typically rely on static benchmarks, golden answers, or automated quality scores, which impose a fixed notion of response quality and overlook pluralistic user preferences in real-world interactions. We propose **RouteJudge**, an online pairwise preference evaluation framework for LLM routing systems, with a public platform available at **routejudge.cn**. RouteJudge evaluates router-level decision quality rather than model-level response quality. For each user query, multiple routing strategies recommend candidate models, and the corresponding model outputs are presented to users through anonymous pairwise comparisons. User preferences are then attributed to the routing strategies behind the compared responses, together with query, routing decision, response, preference, cost, and latency information. RouteJudge supports routing-oriented analyses, including preference win rate, cost-quality trade-off, task-conditioned performance, pairwise router comparison, and routing behavior diagnostics. By grounding evaluation in pluralistic user preferences rather than fixed golden answers, RouteJudge provides a practical basis for studying preference-aware, cost-aware, and context-adaptive LLM routing.

1. Introduction

The rapid growth of the large language model (LLM) ecosystem has made model selection a central component of practical LLM applications (Achiam et al., 2023; Team et al., 2023). Modern LLMs differ substantially in reasoning ability, inference cost, latency, context length, multimodal capability, safety behavior, and response style. Consequently, many systems no longer rely on a single fixed model, but instead employ LLM routing mechanisms to select a suitable model for each incoming query (Shnitzer et al., 2023; Hu et al., 2024; Ding et al., 2024; Ong et al., 2025). The goal of routing is therefore not simply to choose the strongest model, but to make context-dependent decisions that balance response quality, cost, latency, and user satisfaction (Chen et al., 2023; Zhang et al., 2023; Šakota et al., 2024).

Despite this growing importance, the evaluation of LLM routing systems remains largely tied to offline protocols. Existing routing methods are commonly assessed using static benchmarks, golden answers, automated scores, or routing accuracy against an assumed best model (Hu et al., 2024; Huang et al., 2025). Such protocols are convenient and reproducible, but they implicitly reduce response quality to a fixed objective target. This abstraction is useful for controlled comparison, yet it provides only a partial view of routing performance in application-oriented settings.

A central challenge is that many real-world LLM queries do not admit a single universally optimal response. For tasks such as creative writing, translation, tutoring, analytical reasoning, and dialogue, multiple outputs may be factually valid, but users may still prefer different responses depending on their expectations and constraints (Kirk et al., 2024; Sorensen et al., 2024; Feng et al., 2024). Some users favor concise and direct answers, while others prefer detailed explanations; some are sensitive to cost or latency, while others prioritize maximum quality; some value rigorous reasoning, while others care more about tone, fluency, or stylistic fit. Consequently, a router that performs well under golden-answer-based or automated evaluation may not necessarily select models whose responses are preferred by users in deployment.

We frame this mismatch as a pluralistic preference alignment problem for LLM routing. Instead of assuming a single fixed notion of response quality, routing evaluation should account for heterogeneous users, diverse task contexts, and different trade-offs among quality, cost, and latency. Under this view, a routing strategy should be evaluated not only by whether it selects a benchmark-optimal model, but also by whether its decisions lead to responses that users actually

[1]School of Artificial Intelligence, Nanjing University, China [2]National Key Laboratory for Novel Software Technology, Nanjing University, China. Correspondence to: Han-Jia Ye <yehj@lamda.nju.edu.cn>.

Pluralistic Alignment Workshop @ ICML 2026, Seoul, South Korea.

prefer under realistic query distributions.

To address this limitation, we propose **RouteJudge**, an online pairwise preference evaluation framework for LLM routing systems. RouteJudge is designed to evaluate routing decisions under pluralistic user preferences, rather than measuring agreement with a fixed golden answer or an assumed best model. For each user query, multiple routing strategies independently select candidate models under the same model pool and deployment constraints. The selected models generate responses that are presented to users through **anonymous pairwise comparisons**. The resulting preference feedback is then attributed to the routing strategies that produced the compared decisions, enabling RouteJudge to measure whether different routers select models whose responses are preferred by users.

RouteJudge stores each evaluation instance as a routing-centered record, including the user query, optional context and multimodal input, routing decisions, paired model responses, presentation order, preference label, inference cost, latency, task type, and metadata. This structure supports preference-aware and cost-aware routing evaluation beyond aggregate routing accuracy, including preference win rate, Elo rating, cost-quality Pareto analysis, task-conditioned performance, pairwise router comparison, participation rate, and routing behavior diagnostics. The main contributions of this paper are summarized as follows:

- We formulate LLM routing evaluation as a pluralistic preference alignment problem, highlighting the limitations of static golden-answer evaluation for open-ended, application-oriented, and user-dependent tasks.
- We introduce **RouteJudge**, an online routing evaluation protocol based on anonymous pairwise comparison, human preference feedback, preference attribution, and cost recording. RouteJudge shifts routing evaluation from fixed golden-answer supervision toward pluralistic user preference feedback.
- We propose a set of metrics for preference-aware, cost-aware, and task-conditioned routing evaluation, including preference win rate, Elo rating, cost-quality Pareto analysis, task-conditioned performance, pairwise router comparison, participation rate, and routing behavior diagnostics.

2. Background and Motivation

2.1. LLM Routing

LLM routing aims to coordinate a pool of language models with different capabilities, costs, and response characteristics. Given a user query and its surrounding context, a router selects one or more candidate models to answer the query, with the goal of improving the trade-off among response quality, inference cost, latency, and task-specific requirements (Shnitzer et al., 2023; Lai & Ye, 2026). Compared with always using a single fixed model, routing provides a more flexible inference paradigm: easier queries can be assigned to cheaper or faster models, while more difficult or preference-sensitive queries can be assigned to stronger models (Varangot-Reille et al., 2025; Lai et al., 2026).

Existing routing methods instantiate this idea in different ways. Similarity-based routers estimate model suitability by comparing the current query with previously evaluated examples in an embedding space (Zhuang et al., 2024; Reimers & Gurevych, 2019). Learned cost-quality routers train predictive models to estimate the expected utility of each candidate model under a budget constraint (Ding et al., 2024; Ong et al., 2025; Šakota et al., 2024). Cascade and uncertainty-based routers first query a cheaper model and invoke a stronger model only when confidence is low or the initial response appears insufficient (Jiang et al., 2023; Aggarwal et al., 2024). Preference-based and structured routers further incorporate human or model preference signals, graph relations, or ensemble-style decisions to improve model selection (Ramírez et al., 2024; Yue et al., 2023).

Although these methods differ in routing mechanism, they usually share a common evaluation assumption: routing quality can be measured offline using benchmark labels, task-specific metrics, or automated judges (Hu et al., 2024; Huang et al., 2025; Ma et al., 2026). Under this protocol, a router is considered effective if the model it selects obtains a high score on a fixed benchmark. This assumption is convenient for reproducible comparison, but it also narrows the evaluation target. It measures whether a router selects the benchmark-optimal model, rather than whether the routing decision matches what real users would prefer under heterogeneous quality, cost, and latency expectations.

2.2. Why Offline Routing Evaluation Is Insufficient

Offline evaluation provides a controlled and reproducible basis for comparing routing methods, but it offers only a partial characterization of routing quality in realistic user-facing settings. The key limitation is that offline protocols typically reduce routing evaluation to agreement with fixed labels, static benchmark scores, or an assumed best model. However, a router is not merely a response scorer: it makes a deployment decision about which model should answer a query under a particular user objective, task context, and resource constraint.

This mismatch becomes evident in open-ended generation tasks. In writing, dialogue, translation, summarization, tutoring, and analytical reasoning, multiple responses may be factually valid while differing in tone, structure, level of detail, reasoning style, and presentation. Automatic metrics and benchmark labels can capture certain aspects of quality, but they cannot fully determine which response is prefer-

able in a specific interaction (Zhang et al., 2019; Bevilacqua et al., 2025). Consequently, a router that performs well under offline correctness or automated scoring may still select a model whose output is less preferred by users.

A second mismatch arises from preference heterogeneity. User preferences are not governed by a single universal quality function: some users prefer concise answers, while others prefer detailed explanations; some prioritize low cost or low latency, while others value stronger reasoning, formatting, or stylistic fit (Kirk et al., 2024; Sorensen et al., 2024; Feng et al., 2024). Thus, the same query may reasonably call for different routing decisions depending on the user and application scenario. Evaluating a router against a fixed benchmark-optimal model obscures this pluralistic nature of preference.

A third mismatch concerns the objective of routing itself. In deployment, routing is a multi-objective decision over quality, cost, and latency, whereas offline evaluation often treats quality as the primary score and reports cost only as an auxiliary statistic (Chen et al., 2023; Ding et al., 2024). A routing decision is useful only when its quality improvement justifies the additional cost or latency for the target user or application. Static benchmark scores cannot directly capture this preference-conditioned trade-off.

These limitations suggest that routing evaluation should move beyond asking whether a selected model matches an offline target. Instead, it should assess whether the selected model produces responses that users actually prefer under realistic query distributions, heterogeneous preferences, and deployment constraints. **RouteJudge** is designed around this principle: it collects blinded pairwise preference feedback for model outputs selected by different routers and attributes the resulting preferences back to the corresponding routing decisions. In this way, the evaluation target shifts from fixed-target offline scoring to preference-aware assessment of router-level decision quality.

3. Problem Formulation

3.1. LLM Routing

Let $\mathcal{M} = \{m_1, m_2, \ldots, m_K\}$ denote a pool of candidate language models. Given a user query x, optional multi-turn conversation history h, and optional multimodal input I such as an image, LLM routing aims to select a suitable model from $\mathcal{M}$ under deployment constraints. Here, h may be empty for single-turn interactions, and I may be absent for text-only queries.

In practical deployment, users or applications may specify a cost budget C. Let $\hat{c}(m \mid x, h, I)$ denote the estimated inference cost of model m for the current input. The budget-feasible model set is defined as

$$\mathcal{M}_C(x, h, I) = \{m \in \mathcal{M} \mid \hat{c}(m \mid x, h, I) \le C\}.$$

If no budget is specified, we set $C = \infty$ and have $\mathcal{M}_C(x, h, I) = \mathcal{M}$.

A routing strategy $r(\cdot)$ takes the available query context and the budget-feasible model set as input, and outputs one recommended model:

$$r(x, h, I, \mathcal{M}_C) = m^\star, \quad m^\star \in \mathcal{M}_C.$$

Ideally, $m^\star$ should maximize the user-facing utility of the response under the given query, context, and constraints. This utility may depend on response quality, inference cost, latency, task type, and user preference. However, such utility is usually not directly observable before deployment.

Existing routing evaluations usually rely on offline utility functions that combine a fixed benchmark quality score with inference cost. For a router r, let $m_r = r(x, h, I, \mathcal{M}_C)$ denote its selected model. A common offline objective can be written as

$$U_\lambda(r) = \mathbb{E}_{(x,h,I)\sim\mathcal{D}}\left[q_{\text{off}}(x, h, I, m_r) - \lambda \hat{c}(m_r \mid x, h, I)\right],$$

where $q_{\text{off}}(\cdot)$ is an offline quality score, $\hat{c}(\cdot)$ is the estimated inference cost, and λ controls the quality–cost trade-off. By varying λ, one can obtain a cost–quality trade-off curve and compare routers using statistics such as the best utility, average utility, or area under the curve. While convenient for controlled benchmarking, this evaluation still depends on a fixed offline quality function and therefore cannot directly capture heterogeneous user preferences.

3.2. Pluralistic User Preference Evaluation

RouteJudge evaluates routing strategies through online user preference feedback. Let $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$ denote the set of routing strategies to be evaluated. For each evaluation instance, the user provides a query x, optional conversation history h, optional multimodal input I, and an optional cost budget C. RouteJudge first estimates the inference cost of each candidate model and constructs the budget-feasible model set $\mathcal{M}_C(x, h, I)$.

Each routing strategy independently selects one model from the feasible set. We denote the decision of router r_i as

$$m_i = r_i(x, h, I, \mathcal{M}_C), \quad i = 1, \ldots, N.$$

RouteJudge then aggregates these routing decisions into model-level votes. For each model $m \in \mathcal{M}_C$, its vote count is

$$v(m) = \left|\{i \mid m_i = m\}\right|.$$

By default, RouteJudge selects the two models with the highest vote counts for pairwise comparison. If there is a

tie, the model with fewer historical comparison records is prioritized to improve evaluation coverage. If the tie remains unresolved, RouteJudge breaks the tie randomly.

Let the two selected models be m_A and m_B. Their responses are generated independently and presented to the user in anonymous order. The user then provides one of four preference labels:

$$y \in \{\texttt{A Win}, \texttt{B Win}, \texttt{Tie}, \texttt{Both Bad}\}.$$

The label reflects the user's preference between the two displayed responses without revealing the identities of the underlying models or routing strategies.

Each RouteJudge record contains the user query, optional history, optional multimodal input, cost budget, feasible model set, router decisions, model vote counts, selected model pair, anonymous presentation order, paired responses, user preference label, attributed router scores, inference cost, latency, task type, and metadata. This routing-centered record supports preference-aware and cost-aware analyses, including preference win rate, Elo rating, cost-quality Pareto analysis, task-conditioned performance, pairwise router comparison, participation rate, and routing behavior diagnostics.

4. The RouteJudge Evaluation Framework

RouteJudge evaluates LLM routing strategies through online pairwise preference comparison. Unlike conventional benchmarks that store queries and fixed reference answers, each RouteJudge sample records the complete routing decision process, including the user-selected budget, budget-constrained router recommendations, paired model outputs, user preference feedback, and cost-related information. This design allows RouteJudge to evaluate whether a routing strategy selects models whose responses are preferred by users under realistic deployment constraints.

In the current implementation, RouteJudge contains 19 routing strategies and 17 candidate models. The routing strategies cover rule-based, regression-based, classification-based, ranking-based, and non-parametric routers. The model pool includes both high-capability frontier models and lower-cost alternatives. All routing strategies operate over the same shared model pool, while their feasible choices are constrained by the cost budget selected by the user. The full list of routing strategies and candidate models is provided in Appendix A.

4.1. Evaluation Record

Each evaluation sample in RouteJudge is represented as a routing-centered decision record. Formally, a record can be written as

$$\mathcal{Z} = (x, h, I, C, \mathcal{M}_C, \mathbf{d}, \mathbf{v}, m_A, m_B, y, \mathbf{s}, \mathbf{c}, \boldsymbol{\ell}, \tau, \eta)\,,$$

where x is the user query, h is the optional conversation history, I is the optional multimodal input, C is the user-selected cost budget, and $\mathcal{M}_C$ is the budget-feasible model set. The vector $\mathbf{d}$ stores the decisions made by all routing strategies, and $\mathbf{v}$ records the corresponding model-level vote distribution. The models m_A and m_B denote the selected duel pair shown to the user. The variable y is the user preference label, $\mathbf{s}$ stores the attributed router-level scores, $\mathbf{c}$ and $\boldsymbol{\ell}$ record the actual inference costs and latencies, τ denotes the task label, and η contains additional metadata.

This record structure is designed around routing decisions rather than isolated model responses. It therefore supports analyses of not only which model wins a comparison, but also which routing strategies selected that model, how often each router participates in pairwise comparisons, how performance changes under different budgets, and how routing behavior varies across task types.

4.2. Evaluation Workflow

Figure 1 summarizes the RouteJudge evaluation pipeline, from query submission and budget selection to router-level preference attribution and record construction. For each user query, RouteJudge follows an eight-step workflow.

Step 1: Query and budget submission. The user submits a text-only or multimodal query to the platform and selects a cost budget C. The budget specifies the maximum allowable inference cost for candidate model selection. At this stage, the user cannot observe model identities, router decisions, vote distributions, or cost information.

Step 2: Budget-feasible model filtering. Given the user-selected budget C, RouteJudge constructs a feasible candidate set $\mathcal{M}_C$ from the full model pool $\mathcal{M}$. Only models whose estimated inference cost falls within the selected budget are retained:

$$\mathcal{M}_C = \{m \in \mathcal{M} \mid \hat{c}(m \mid x, h, I) \leq C\}.$$

This step ensures that all subsequent router decisions and pairwise comparisons are made under the same user-specified cost constraint.

Step 3: Router recommendation. The query x, optional history h, optional multimodal input I, budget C, and feasible candidate set $\mathcal{M}_C$ are sent to all routing strategies in parallel. Each router independently recommends one model from $\mathcal{M}_C$. Let $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$ denote the router set. The decision of router r_i is

$$m_i = r_i(x, h, I, \mathcal{M}_C), \quad m_i \in \mathcal{M}_C.$$

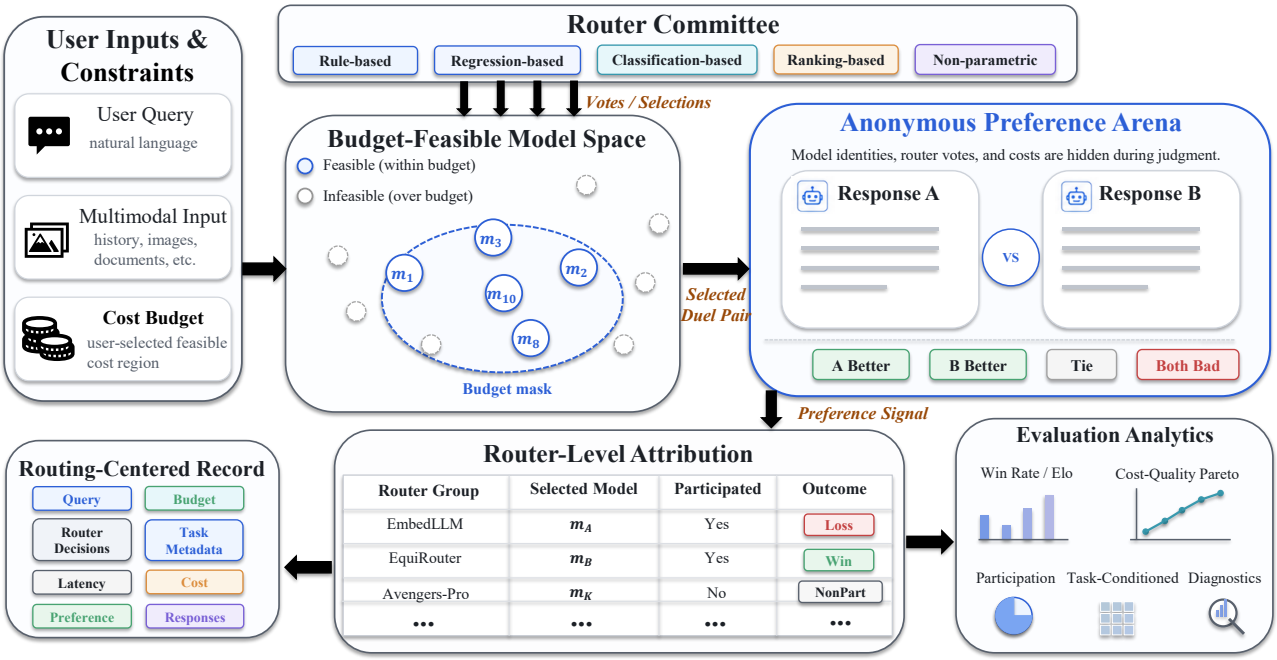


Figure 1. Overview of the RouteJudge evaluation framework. Given a user query, optional multimodal input, and a user-selected cost budget, a committee of routing strategies recommends models from the budget-feasible model space. RouteJudge selects a duel pair, presents the two responses through an anonymous preference interface, and attributes the resulting preference signal back to the routing strategies behind the compared model choices. Each interaction is stored as a routing-centered record containing the query, budget, router decisions, selected responses, preference label, cost, latency, and task metadata, supporting downstream analyses such as win rate, Elo rating, cost-quality Pareto analysis, participation, and task-conditioned diagnostics.

RouteJudge treats each router as a black-box decision maker and does not require different routers to share the same internal scoring function.

Step 4: Duel model selection. RouteJudge aggregates router recommendations into model-level votes. For each model $m \in \mathcal{M}_C$, its vote count is

$$v(m) = |\{i \mid m_i = m\}|.$$

By default, RouteJudge selects the two distinct models with the highest vote counts as the duel pair (m_A, m_B). If there is a tie, the model with fewer historical comparison records is prioritized to improve evaluation coverage. If the tie remains unresolved, RouteJudge breaks the tie randomly.

Step 5: Parallel response generation. The two selected models generate responses to the same user query in parallel. RouteJudge records the actual inference cost and response latency of each model. Parallel generation reduces the risk that differences in waiting time affect the user's preference judgment.

Step 6: Anonymous pairwise presentation. The two responses are shown to the user in randomized anonymous order. Model identities, router decisions, vote counts, and cost information are hidden during preference judgment. This design reduces model-brand bias, router-induced bias, and position bias.

Step 7: User preference judgment. The user selects one of four preference labels:

$$y \in \{\texttt{A Win}, \texttt{B Win}, \texttt{Tie}, \texttt{Both Bad}\}.$$

This four-valued label space avoids forcing a binary choice when the two responses are indistinguishable or both unsatisfactory.

Step 8: Result reveal and router attribution. After the user submits the preference label, RouteJudge reveals the model identities, router vote distribution, actual costs, latencies, and router-level attribution results. The user preference label is first converted into comparison scores:

$$(s_A, s_B) = \begin{cases} (1,0), & y = \texttt{A Win}, \\ (0,1), & y = \texttt{B Win}, \\ (0.5,0.5), & y = \texttt{Tie}, \\ (0,0), & y = \texttt{Both Bad}. \end{cases} \tag{1}$$

The scores are then attributed back to the routing strategies that selected the compared models:

$$S_i = \begin{cases} s_A, & m_i = m_A, \\ s_B, & m_i = m_B, \\ \varnothing, & m_i \notin \{m_A, m_B\}. \end{cases}$$

Here, $\varnothing$ indicates that router r_i does not participate in this comparison and is not counted as either a win or a loss. The complete record is stored for downstream analysis of router performance under different budgets, task types, and user preference patterns.

4.3. Task Metadata

To support task-conditioned analysis, RouteJudge assigns each query a task label τ. In the current implementation, queries are mapped to six broad categories: Coding, Math, Translation, Creative Writing, Analysis, and Other. This metadata allows RouteJudge to report not only overall router performance, but also router behavior across heterogeneous task types. For example, a router may perform well on coding and mathematical reasoning while being less competitive on translation or creative writing. Task-conditioned analysis is therefore necessary for understanding when and why a routing strategy is effective.

4.4. Routing Strategy Training

The routing strategies in RouteJudge can be trained or initialized using existing public benchmarks that cover both text-only and multimodal tasks, including GISA (Zhu et al., 2026), LiveBench (White et al., 2025), OCRBench v2 (Fu et al., 2025), and VisBrowse (Zhang et al., 2026b). These datasets provide query distributions, task annotations, model outputs, and model-performance signals that can be used to develop different types of routers.

Different router families use these data in different ways. Rule-based routers rely on manually designed heuristics, such as task type, estimated difficulty, cost budget, or modality. Regression-based routers learn to predict the expected utility of each candidate model, often using benchmark quality scores and inference costs as supervision. Classification-based routers directly predict the best model for each input according to offline labels. Ranking-based routers learn an ordering over candidate models rather than selecting a single model independently. Non-parametric routers retrieve similar historical examples and infer the model choice from their observed performance.

Importantly, RouteJudge separates router training from router evaluation. Public benchmarks are used to initialize or train routing strategies, while RouteJudge evaluates their decisions through online preference feedback. This separation allows different routers to use different training mechanisms while being compared under the same user-facing evaluation protocol.

4.5. Handling Non-Participating Routers

A router may recommend a model that does not enter the final pairwise comparison. In this case, the router is marked as *Non-participating* for that query and is not counted in the win/loss denominator. This avoids assigning credit or blame to a router whose recommended model was not actually judged by the user.

However, non-participation itself is informative. A router that frequently recommends models outside the selected duel pair may have limited evaluation coverage, while a conservative router that repeatedly recommends popular models may participate more often. RouteJudge therefore reports participation rate alongside preference-based metrics, making it possible to distinguish high win rate from high evaluation coverage.

5. Evaluation Metrics

RouteJudge evaluates routing strategies from four complementary perspectives: preference-based performance, cost–quality trade-off, task-conditioned behavior, and routing decision patterns. Let $\mathcal{D}_{\mathrm{RJ}} = \{\mathcal{Z}_t\}_{t=1}^{T}$ denote the set of RouteJudge evaluation records. For the t-th record, $m_i^{(t)}$ denotes the model selected by router r_i, $(m_A^{(t)}, m_B^{(t)})$ denotes the selected duel pair, $y^{(t)}$ denotes the user preference label, $S_i^{(t)} \in \{0, 0.5, 1, \varnothing\}$ denotes the attributed score of router r_i, and $\tau^{(t)}$ denotes the task label.

5.1. Preference-Based Router Performance

The primary evaluation signal in RouteJudge is the user preference attributed to each router. Since a router may recommend a model that does not enter the final duel pair, we first define the participating set of router r_i as

$$\mathcal{P}_i = \{t \mid S_i^{(t)} \neq \varnothing\}.$$

The participation rate is then

$$\mathrm{PartRate}(r_i) = \frac{|\mathcal{P}_i|}{T}.$$

This metric prevents a router with high preference performance but limited evaluation coverage from being over-interpreted.

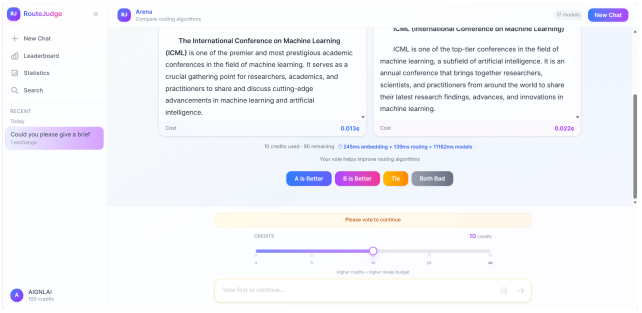
We report two preference-based performance metrics. The first is the average preference score:

$$\mathrm{PrefScore}(r_i) = \frac{1}{|\mathcal{P}_i|} \sum_{t \in \mathcal{P}_i} S_i^{(t)}.$$

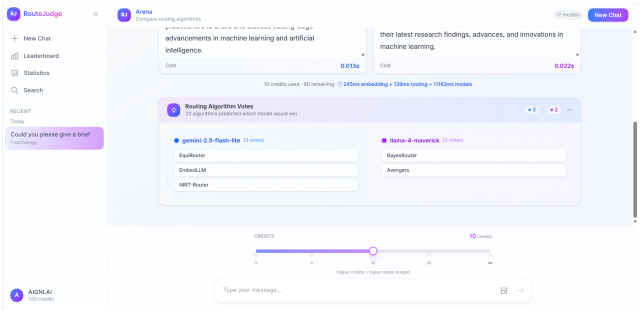
This score uses the attribution rule defined in Equation (1), where wins receive 1, ties receive 0.5, losses receive 0, and *Both Bad* contributes 0 to the participating routers.

The second metric is the binary preference win rate, computed only on comparisons with a clear winner. Let

$$\mathcal{B}_i = \{t \in \mathcal{P}_i \mid y^{(t)} \in \{\texttt{A Win}, \texttt{B Win}\}\}.$$



(a) Anonymous pairwise preference interface in RouteJudge. Two model responses are presented side by side in randomized order, with model identities and router votes hidden during judgment. Users provide one of four labels: A Win, B Win, Tie, or Both Bad.



(b) Result reveal and router attribution interface in RouteJudge. After the user submits a preference label, RouteJudge reveals the model identities, displays the routing vote distribution, reports cost and latency information, and assigns router-level outcomes according to whether each router selected the user-preferred model.

Figure 2. RouteJudge user interface. Left: anonymous pairwise preference interface used for blinded user judgment. Right: result reveal and router attribution interface shown after preference submission.

The win rate is defined as

$$\mathrm{WinRate}(r_i) = \frac{\sum_{t \in \mathcal{B}_i} \mathbf{1}[S_i^{(t)} = 1]}{|\mathcal{B}_i|}.$$

Thus, *Tie* and *Both Bad* are excluded from the binary win/loss denominator, while they are still retained in the evaluation records for tie analysis and failure analysis.

For robustness, we also report Elo ratings and bootstrap confidence intervals. Elo ratings provide a relative ranking of routers based on pairwise preference outcomes, while bootstrap confidence intervals estimate the uncertainty of preference scores and win rates, especially when the number of participating samples is limited.

5.2. Cost–Quality Trade-off

A central goal of LLM routing is to improve user-preferred quality under cost constraints. For router r_i, we compute its average participating cost as

$$\mathrm{Cost}(r_i) = \frac{1}{|\mathcal{P}_i|} \sum_{t \in \mathcal{P}_i} c^{(t)}\left(m_i^{(t)}\right),$$

where $c^{(t)}(m_i^{(t)})$ denotes the actual inference cost of the model selected by r_i in the t-th comparison. We use participating records because only models in the duel pair are actually invoked and judged by the user.

RouteJudge analyzes routers in the cost–quality space, where $\mathrm{PrefScore}(r_i)$ or $\mathrm{WinRate}(r_i)$ serves as the quality axis and $\mathrm{Cost}(r_i)$ serves as the cost axis. By default, we use $\mathrm{PrefScore}(r_i)$ because it preserves non-binary outcomes. A router r_i dominates another router r_j if

$$\mathrm{PrefScore}(r_i) \geq \mathrm{PrefScore}(r_j), \quad \mathrm{Cost}(r_i) \leq \mathrm{Cost}(r_j),$$

with at least one inequality being strict. Routers on the Pareto frontier represent strategies that achieve the best observed preference performance at a given cost level.

When evaluations are grouped by user-selected budget, the same analysis can be performed for each budget level C, yielding a budget-conditioned cost–quality curve. This allows RouteJudge to compare whether a router remains effective under strict, moderate, or loose cost constraints.

5.3. Task-Conditioned Performance

Overall performance may hide substantial variation across task types. RouteJudge therefore reports preference metrics conditioned on the task label τ. For a task category τ, define

$$\mathcal{P}_{i,\tau} = \{t \in \mathcal{P}_i \mid \tau^{(t)} = \tau\}.$$

The task-conditioned preference score is

$$\mathrm{PrefScore}(r_i \mid \tau) = \frac{1}{|\mathcal{P}_{i,\tau}|} \sum_{t \in \mathcal{P}_{i,\tau}} S_i^{(t)}.$$

Similarly, the task-conditioned win rate is computed by restricting the binary winning set $\mathcal{B}_i$ to records with $\tau^{(t)} = \tau$. These metrics reveal whether a router is broadly reliable or specialized for certain query types, such as coding, math, translation, creative writing, or analysis.

5.4. Pairwise Router Comparison

Aggregate scores do not always reveal whether one router consistently outperforms another on the same queries. We therefore compute pairwise router comparisons on shared participating records. For two routers r_i and r_j, define

$$\mathcal{P}_{ij} = \mathcal{P}_i \cap \mathcal{P}_j.$$

The head-to-head preference score of r_i against r_j is

$$\mathrm{H2H}(r_i, r_j) = \frac{1}{|\mathcal{P}_{ij}|} \sum_{t \in \mathcal{P}_{ij}} \phi\Big(S_i^{(t)}, S_j^{(t)}\Big),$$

where

$$\phi(a,b) = \begin{cases} 1, & a > b, \\ 0.5, & a = b, \\ 0, & a < b. \end{cases}$$

This metric directly compares two routers on records where both selected one of the compared models.

We further use McNemar's test to assess whether the difference between two routers is statistically significant on binary preference outcomes. Let a be the number of shared binary comparisons where r_i receives a win and r_j receives a loss, and let b be the number of shared binary comparisons where r_j receives a win and r_i receives a loss. The test statistic is

$$\chi^2 = \frac{(|a-b|-1)^2}{a+b}.$$

The p-value is computed under a chi-squared distribution with one degree of freedom. This test is applied only when $a + b > 0$ and the user preference label has a clear winner.

5.5. Routing Behavior Diagnostics

Beyond ranking routers, RouteJudge also analyzes how routing strategies make decisions. These diagnostics help distinguish genuinely effective routers from routers that achieve high scores through narrow coverage, excessive conservatism, or redundant decision patterns.

Routing entropy. Let $p_{i,m}$ denote the empirical frequency with which router r_i selects model m across all evaluation records:

$$p_{i,m} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[m_i^{(t)} = m].$$

The selection entropy of router r_i is

$$H(r_i) = -\sum_{m \in \mathcal{M}} p_{i,m} \log_2 p_{i,m},$$

and the normalized entropy is

$$H_{\mathrm{norm}}(r_i) = \frac{H(r_i)}{\log_2 |\mathcal{M}|}.$$

Low entropy indicates that a router repeatedly selects a small subset of models, while high entropy indicates more diverse routing behavior.

Router agreement. For two routers r_i and r_j, the raw agreement rate is

$$\mathrm{Agree}(r_i, r_j) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[m_i^{(t)} = m_j^{(t)}].$$

We also report Cohen's Kappa:

$$\kappa(r_i, r_j) = \frac{P_o - P_e}{1 - P_e},$$

where $P_o = \mathrm{Agree}(r_i, r_j)$ and

$$P_e = \sum_{m \in \mathcal{M}} p_{i,m} p_{j,m}.$$

High agreement may indicate redundant routing behavior, while low agreement suggests that routers rely on different decision rules or specialize in different regions of the query space.

Consensus strength. For each evaluation record, RouteJudge aggregates router decisions into model-level votes $v^{(t)}(m)$. The consensus strength is defined as

$$\rho^{(t)} = \frac{\max_{m \in \mathcal{M}_C^{(t)}} v^{(t)}(m)}{N},$$

where N is the number of routing strategies. A high value of $\rho^{(t)}$ indicates that many routers select the same model, while a low value indicates disagreement among routers. RouteJudge analyzes whether higher consensus strength correlates with higher user preference outcomes, thereby testing whether router agreement is a reliable signal of response preference.

6. Results and Discussion

We report preliminary results collected from the RouteJudge platform as of **2026-06-08 04:00 AoE** (UTC-12). As of that time, the platform had recorded 261 matches and 109 user votes. The following analysis is based on the 109 voted comparisons. Since the number of votes is still limited, the results should be interpreted as an initial empirical snapshot rather than a definitive ranking of routing strategies. Additional analyses are provided in **Appendix B**, and continuously updated platform statistics are available at **https://routejudge.cn/stats**.

6.1. Router Ranking

Table 1 reports the current router ranking by Elo score. Overall, the leading routers achieve both higher Elo ratings and above-random preference win rates, suggesting that online user preferences can already differentiate routing strategies even at this early stage. RouterLLM-MF obtains the highest Elo score, while NIRT-Router achieves the highest observed win rate. Several embedding-based, graph-based, and regression-style routers also remain competitive, indicating that different routing paradigms can be effective under user-preference-based evaluation.

At the same time, the ranking should be read together with the number of available comparisons and the participation

Table 1. Router performance ranked by Elo score. Results are based on 109 user-voted comparisons collected by RouteJudge.

Router	Win Rate	Elo
RouterLLM-MF (Ong et al., 2025)	66.67%	**1278**
NIRT-Router (Song et al., 2025)	**80.00%**	1274
kNNRouter (Stripelis et al., 2024)	60.00%	1240
GraphRouter (Feng et al., 2025)	44.44%	1218
EmbedLLM (Zhuang et al., 2024)	63.64%	1215
EquiRouter (Lai & Ye, 2026)	64.00%	1212
RouterDC (Chen et al., 2024)	57.14%	1209
SVMRouter (Stripelis et al., 2024)	51.72%	1205
Avengers-Pro (Zhang et al., 2025)	50.00%	1204
Avengers (Zhang et al., 2026a)	58.33%	1198
RM-Interval (Tsiourvas et al., 2025)	46.15%	1184
MLPRouter (Stripelis et al., 2024)	39.13%	1176
Eagle (Zhao et al., 2024)	43.48%	1175
OmniRouter (Mei et al., 2025)	40.00%	1170
RM-CLS (Tsiourvas et al., 2025)	46.43%	1169
RouteLLM-SW (Ong et al., 2025)	27.27%	1145
RM-Softmax (Tsiourvas et al., 2025)	31.82%	1136
HybridLLM (Ding et al., 2024)	18.18%	1134
MIRT-Router (Song et al., 2025)	36.00%	1117

behavior of each router. A router with a high win rate may have participated in fewer decisive comparisons, while a router with a lower win rate may have been evaluated under more diverse or more difficult query conditions. This is one reason why RouteJudge reports Elo, win rate, participation, and cost-related statistics jointly rather than relying on a single aggregate metric.

The preliminary ranking also shows that strong offline routing designs do not necessarily transfer uniformly to online preference-based evaluation. Some routers with explicit learned scoring mechanisms obtain relatively low preference win rates, while simpler non-parametric or matrix-factorization-based approaches remain competitive. This observation supports the motivation of RouteJudge: router quality should not be assessed only by agreement with offline labels or static benchmark scores, but also by whether the selected models produce responses preferred by users under real interaction settings.

6.2. Model Pareto Frontier

We further examine the model-level cost–preference trade-off in RouteJudge. Figure 3 plots each candidate model by its average inference cost and empirical preference win rate. The dashed line denotes the observed Pareto frontier, where no model is simultaneously cheaper and preferred more often.

The results show that model preference is not determined by cost alone. Some low-cost models achieve competitive win rates, while several stronger models require substantially higher budgets. This suggests that effective routing should adapt model selection to the user-selected budget and task context, rather than always choosing the most expensive or highest-capability model.

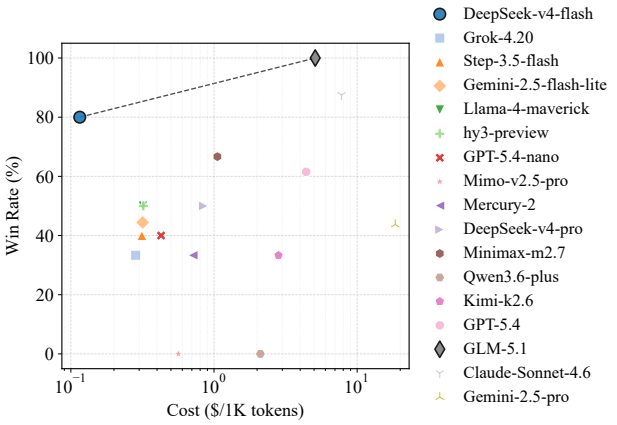


Figure 3. Model-level cost–win-rate distribution on RouteJudge. Each point represents a candidate model, with cost measured in dollars per 1K tokens and win rate computed from user-voted comparisons. The dashed line marks the observed Pareto frontier.

7. Conclusion

This paper introduces RouteJudge, an online pairwise preference evaluation framework for LLM routing systems. Instead of evaluating routers only through fixed benchmark labels or automated scores, RouteJudge measures whether routing decisions lead to responses preferred by users under budget, cost, latency, and task constraints. The framework records the full routing decision process, attributes anonymous pairwise preference feedback to routing strategies, and supports preference-aware, cost-aware, and task-conditioned analyses. Overall, the leading routers obtain higher Elo ratings and above-random preference win rates, suggesting that online user preferences may provide a useful signal for differentiating routing strategies even at this early stage.

Acknowledgements

This work was supported by the National Key R&D Program of China under Grant 2024YFE0202800.

Impact Statement

This work aims to improve the evaluation of LLM routing systems by grounding router assessment in pluralistic user preferences rather than fixed benchmark targets. By enabling preference-aware and cost-aware analysis, RouteJudge may help researchers and practitioners better understand how routing decisions affect user satisfaction, affordability, and deployment efficiency. We explicitly account for privacy-related, fairness, and misuse risks in the design of RouteJudge: model identities, router decisions, vote distribu-

tions, and cost information are hidden during user judgment to reduce brand and presentation bias; preference feedback is attributed at the router level rather than used as a universal measure of model quality; and the collected records preserve metadata needed to analyze task-conditioned behavior, participation imbalance, and potential feedback bias. These design choices help make online routing evaluation more transparent and preference-aware, while discouraging the interpretation of aggregated preference rankings as absolute or universally valid measures of router quality.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Aggarwal, P., Madaan, A., Anand, A., Potharaju, S. P., Mishra, S., Zhou, P., Gupta, A., Rajagopal, D., Kappaganthu, K., Yang, Y., et al. Automix: Automatically mixing language models. In *NeurIPS*, 2024.

Bevilacqua, M., Oketch, K., Qin, R., Stamey, W., Zhang, X., Gan, Y., Yang, K., and Abbasi, A. When automated assessment meets automated content generation: Examining text quality in the era of gpts. *ACM Transactions on Information Systems*, 2025.

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

Chen, S., Jiang, W., Lin, B., Kwok, J., and Zhang, Y. Routerdc: Query-based router by dual contrastive learning for assembling large language models. In *NeurIPS*, 2024.

Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Rühle, V., Lakshmanan, L. V. S., and Awadallah, A. H. Hybrid LLM: Cost-efficient and quality-aware query routing. In *ICLR*, 2024.

Feng, S., Sorensen, T., Liu, Y., Fisher, J., Park, C. Y., Choi, Y., and Tsvetkov, Y. Modular pluralism: Pluralistic alignment via multi-llm collaboration. In *EMNLP*, 2024.

Feng, T., Shen, Y., and You, J. Graphrouter: A graph-based router for llm selections. In *ICLR*, 2025.

Fu, L., Kuang, Z., Song, J., Huang, M., Yang, B., Li, Y., Zhu, L., Luo, Q., Wang, X., Lu, H., Li, Z., Tang, G., Shan, B., Lin, C., Liu, Q., Wu, B., Feng, H., Liu, H., Huang, C., Tang, J., Chen, W., Jin, L., Liu, Y., and Bai, X. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. In *NeurIPS*, 2025.

Hu, Q. J., Bieker, J., Li, X., Jiang, N., Keigwin, B., Ranganath, G., Keutzer, K., and Upadhyay, S. K. Routerbench: A benchmark for multi-LLM routing system. In *Agentic Markets Workshop at ICML 2024*, 2024.

Huang, Z., Ling, G., Lin, Y., Chen, Y., Zhong, S., Wu, H., and Lin, L. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. In *EMNLP*, 2025.

Jiang, D., Ren, X., and Lin, B. Y. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *ACL*, 2023.

Kirk, H. R., Whitefield, A., Röttger, P., Bean, A., Margatina, K., Ciro, J., Mosquera, R., Bartolo, M., Williams, A., He, H., et al. The prism alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *NeurIPS*, 2024.

Lai, G. and Ye, H.-J. When routing collapses: On the degenerate convergence of llm routers. *arXiv preprint arXiv:2602.03478*, 2026.

Lai, G., Hu, H., Chen, L., Li, Z., and Ye, H.-J. From sampled outcomes to capability distributions: Rethinking supervision for llm routing. *arXiv preprint arXiv:2606.06924*, 2026.

Ma, H., Lai, G., and Ye, H.-J. Mmr-bench: A comprehensive benchmark for multimodal llm routing. *arXiv preprint arXiv:2601.17814*, 2026.

Mei, K., Xu, W., Lin, S., and Zhang, Y. Omnirouter: Budget and performance controllable multi-llm routing. *arXiv preprint arXiv:2502.20576*, 2025.

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., and Stoica, I. Routellm: Learning to route llms from preference data. In *ICLR*, 2025.

Ramírez, G., Birch, A., and Titov, I. Optimising calls to large language models with uncertainty-based two-tier selection. *arXiv preprint arXiv:2405.02134*, 2024.

Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP*, 2019.

Šakota, M., Peyrard, M., and West, R. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 606–615, 2024.

Shnitzer, T., Ou, A., Silva, M., Soule, K., Sun, Y., Solomon, J., Thompson, N., and Yurochkin, M. Llm routing with

benchmark datasets. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.

Song, W., Huang, Z., Cheng, C., Gao, W., Xu, B., Zhao, G., Wang, F., and Wu, R. IRT-router: Effective and interpretable multi-LLM routing via item response theory. In *ACL*, 2025.

Sorensen, T., Jiang, L., Hwang, J. D., Levine, S., Pyatkin, V., West, P., Dziri, N., Lu, X., Rao, K., Bhagavatula, C., et al. Value kaleidoscope: Engaging ai with pluralistic human values, rights, and duties. In *AAAI*, 2024.

Stripelis, D., Hu, Z., Zhang, J., Xu, Z., Shah, A. D., Jin, H., Yao, Y., Avestimehr, S., and He, C. Tensoropera router: A multi-model router for efficient llm inference. In *EMNLP*, 2024.

Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Tsiourvas, A., Sun, W., and Perakis, G. Causal llm routing: End-to-end regret minimization from observational data. In *NeurIPS*, 2025.

Varangot-Reille, C., Bouvard, C., Gourru, A., Ciancone, M., Schaeffer, M., and Jacquenet, F. Doing more with less–implementing routing strategies in large language model-based systems: An extended survey. *arXiv preprint arXiv:2502.00409*, 2025.

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Naidu, S., Hegde, C., LeCun, Y., Goldstein, T., Neiswanger, W., and Goldblum, M. Livebench: A challenging, contamination-limited llm benchmark. In *ICLR*, 2025.

Yue, M., Zhao, J., Zhang, M., Du, L., and Yao, Z. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *arXiv preprint arXiv:2310.03094*, 2023.

Zhang, J., Krishna, R., Awadallah, A. H., and Wang, C. Ecoassistant: Using llm assistant more affordably and accurately. *arXiv preprint arXiv:2310.03046*, 2023.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Zhang, Y., Li, H., Chen, J., Zhang, H., Ye, P., Bai, L., and Hu, S. Beyond gpt-5: Making llms cheaper and better via performance-efficiency optimized routing. In *DAI*, 2025.

Zhang, Y., Li, H., Wang, C., Chen, L., Zhang, Q., Ye, P., Feng, S., Wang, X., Xu, J., Bai, L., et al. The avengers: A routing recipe for collective intelligence in language models. In *AAAI*, 2026a.

Zhang, Z., Su, J., Zhou, Z., Miao, C., Hong, Y., Wu, Q., Liu, Y., Wu, F., Tian, Y., Liang, Y., Shan, Z., Xia, W., Zhang, Y.-F., Zhang, B., Li, Z., Xiang, S., and Yan, Y. Visbrowse-bench: Benchmarking visual-native search for multimodal browsing agents. *arXiv preprint arXiv:2603.16289*, 2026b.

Zhao, Z., Jin, S., and Mao, Z. M. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*, 2024.

Zhu, Y., Zhang, X., Zhang, M., Jin, J., Zhang, L., Song, X., Zhao, K., Zeng, W., Tang, R., Li, H., Wen, J.-R., and Dou, Z. Gisa: A benchmark for general information-seeking assistant. *arXiv preprint arXiv:2602.08543*, 2026.

Zhuang, R., Wu, T., Wen, Z., Li, A., Jiao, J., and Ramchandran, K. Embedllm: Learning compact representations of large language models. In *ICLR*, 2024.

A. Details of Routers and Candidate Models

This appendix provides additional details about the candidate model pool and routing strategies used in RouteJudge. In the current implementation, RouteJudge evaluates routing strategies over a shared pool of 17 candidate models. All routers operate under the same budget-feasible model space for each query, ensuring that performance differences come from routing decisions rather than from access to different model sets.

A.1. Candidate Models

The current RouteJudge platform includes the following 17 candidate models:

- **DeepSeek-v4-flash**: A low-cost candidate model used for budget-sensitive routing decisions.
- **Grok-4.20**: A general-purpose candidate model included in the shared model pool.
- **Step-3.5-flash**: A lightweight candidate model used for efficient response generation.
- **Gemini-2.5-flash-lite**: A low-cost Gemini-family model included for cost-efficient routing.
- **Llama-4-maverick**: An open-family candidate model used as part of the heterogeneous model pool.
- **hy3-preview**: A preview candidate model included to increase model diversity.
- **GPT-5.4-nano**: A compact GPT-family model used for low-cost query handling.
- **Mimo-v2.5-pro**: A pro-level candidate model included for stronger response generation.
- **Mercury-2**: A general candidate model included in the RouteJudge model pool.
- **DeepSeek-v4-pro**: A stronger DeepSeek-family model used for higher-quality routing choices.
- **Minimax-m2.7**: A general-purpose candidate model included for comparison across model families.
- **Qwen3.6-plus**: A Qwen-family model used as a higher-capability candidate in the model pool.
- **Kimi-k2.6**: A Kimi-family candidate model included for diverse generation behavior.
- **GPT-5.4**: A high-capability GPT-family model used for quality-oriented routing decisions.
- **GLM-5.1**: A GLM-family candidate model included in the shared model pool.
- **Claude-Sonnet-4.6**: A high-capability Claude-family model used for quality-sensitive queries.
- **Gemini-2.5-pro**: A pro-level Gemini-family model included for high-quality response generation.

A.2. Routing Strategies

RouteJudge includes routing strategies from multiple families, including rule-based, regression-based, classification-based, ranking-based, graph-based, reward-model-based, cascade-based, and non-parametric methods. The current implementation contains the following routing strategies:

- **RouterLLM-MF** (Ong et al., 2025): A matrix-factorization-based RouteLLM variant that estimates model suitability from observed routing data.
- **RouteLLM-SW** (Ong et al., 2025): A RouteLLM variant that uses similarity- or score-weighted information for model selection.
- **NIRT-Router** (Song et al., 2025): An item-response-theory-based router that models query difficulty and model ability.
- **MIRT-Router** (Song et al., 2025): A multidimensional IRT-based router that represents model ability and query difficulty with multiple latent factors.
- **kNNRouter** (Stripelis et al., 2024): A non-parametric router that selects models according to nearest historical examples.
- **SVMRouter** (Stripelis et al., 2024): A classification-based router that predicts a suitable model from query features.
- **MLPRouter** (Stripelis et al., 2024): A neural classification-based router that maps query representations to model choices.
- **GraphRouter** (Feng et al., 2025): A graph-based router that exploits structured relations among models, tasks, or queries.
- **EmbedLLM** (Zhuang et al., 2024): An embedding-based router that estimates model suitability by comparing queries in representation space.
- **EquiRouter** (Lai & Ye, 2026): A routing strategy designed to select models by estimating relative model utility under routing constraints.
- **RouterDC** (Chen et al., 2024): A data-driven router that learns model-selection decisions from offline routing signals.

- **Avengers** (Zhang et al., 2026a): An ensemble-style router that combines multiple decision signals for model selection.
- **Avengers-Pro** (Zhang et al., 2025): An enhanced Avengers-style router with stronger aggregation or decision mechanisms.
- **Eagle** (Zhao et al., 2024): A cascade-style router that decides whether to invoke stronger models based on estimated difficulty or uncertainty.
- **OmniRouter** (Mei et al., 2025): A general-purpose router designed for model selection across heterogeneous task types.
- **HybridLLM** (Ding et al., 2024): A hybrid cost–quality-aware router that combines multiple routing criteria.
- **RM-CLS** (Tsiourvas et al., 2025): A reward-model-based classification router that predicts the preferred model.
- **RM-Softmax** (Tsiourvas et al., 2025): A reward-model-based router that selects models according to softmax-normalized preference scores.
- **RM-Interval** (Tsiourvas et al., 2025): A reward-model-based router that incorporates interval-style uncertainty in routing decisions.

B. Additional Results

This appendix provides additional analyses of routing behavior in RouteJudge. We focus on two complementary aspects: how different routers distribute their selections over the candidate model pool, and how routers compare against each other under shared user-voted comparisons.

B.1. Model Selection Distribution

Figure A.1 shows the model selection distribution of each routing strategy. Each horizontal bar represents one router, and each segment indicates the fraction of queries for which the router selects a specific candidate model. The distribution reveals substantial differences in routing behavior. Some routers concentrate heavily on a small number of models, indicating more conservative or model-specific decision patterns. Other routers spread their selections across a broader range of models, suggesting stronger sensitivity to query-level differences or budget constraints.

This analysis complements the aggregate router ranking in the main text. A high-ranking router may obtain strong preference performance either by consistently selecting a robust model or by adapting its selections across different queries. Conversely, a router with diverse selections is not necessarily better unless this diversity leads to preferred responses. Therefore, model selection distribution provides a useful diagnostic for interpreting why routers obtain different win rates and Elo scores.

B.2. Head-to-Head Router Comparison

Figure A.2 presents the head-to-head win-rate matrix between routing strategies. Each cell reports the win rate of the row router against the column router on shared comparisons. Unlike global Elo ranking, this matrix directly shows pairwise advantages and disadvantages between routers. This is useful because two routers with similar overall scores may still behave differently on overlapping query subsets.

The heatmap shows that router performance is not uniformly ordered across all pairwise comparisons. Some routers exhibit broad advantages against many alternatives, while others perform competitively only against specific groups of routers. This suggests that RouteJudge can support a more fine-grained analysis than a single leaderboard score: it can reveal whether a router is consistently strong, specialized, or mainly competitive under certain comparison settings.

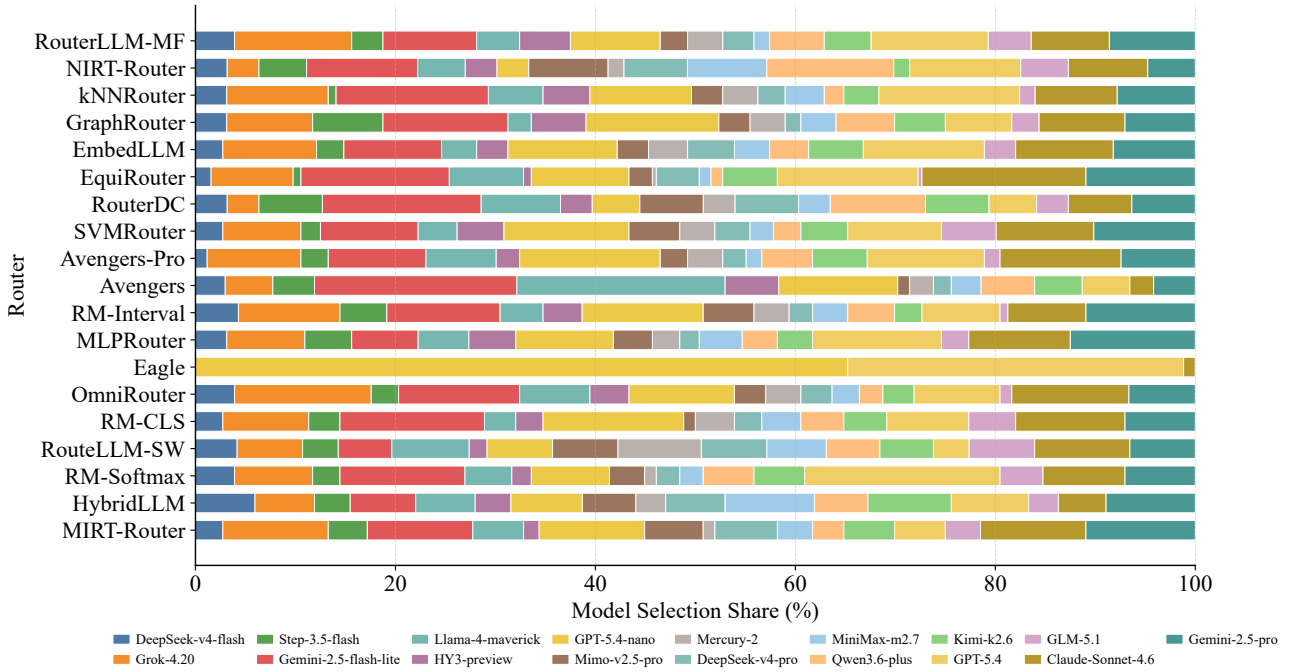


Figure A.1. Model selection distribution of routing strategies. Each horizontal bar shows the percentage of times a router selects each candidate model. Concentrated distributions indicate conservative or model-specific routing behavior, while more diverse distributions suggest broader use of the candidate model pool.

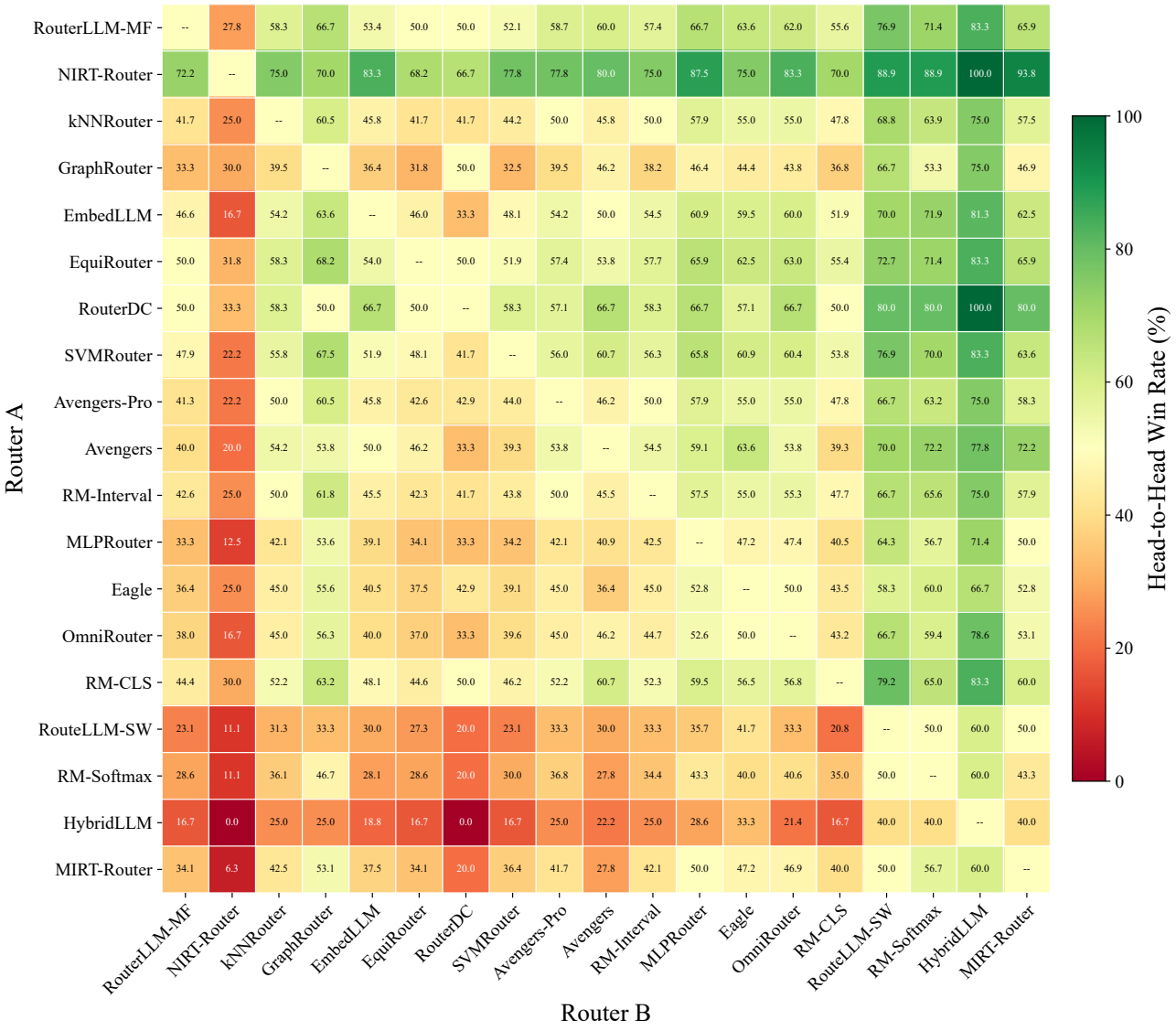


Figure A.2. Head-to-head win-rate heatmap between routing strategies. Each cell reports the percentage of shared comparisons in which the row router outperforms the column router. Higher values indicate stronger pairwise advantage of the row router, while lower values indicate weaker relative performance against the corresponding column router.