Conformal LLM Routing with Distribution-Free Safety Guarantees

Iqtedar Uddin and André Bauer
Illinois Institute of Technology
iuddin1@hawk.illinoistech.edu abauer7@illinoistech.edu

Abstract

LLM routing directs queries to a cheaper model when it suffices and to an expensive model otherwise, reducing inference cost. Existing input-based routers optimize cost-performance trade-offs but provide no formal bound on how often the cheaper model fails among routed queries. We adapt a proactive conformal gate framework to LLM routing. A logistic regression gate trained on text embeddings predicts per-query safety, and Clopper-Pearson conformal calibration selects a routing threshold that guarantees the violation rate among routed queries stays below α (the violation tolerance) with probability at least $1-\delta$ (the confidence level). On two benchmarks covering math reasoning (GSM8K) and multi-domain knowledge (MMLU), routing between Mixtral-8x7B and GPT-4 (a 24.5$\times$ cost difference), our method maintains the target α within the δ tolerance across a sweep from 0.05 to 0.50, while a validation-tuned baseline crosses the violation boundary on GSM8K. A feasibility analysis across all 10 RouterBench models reveals that routability is jointly model- and task-dependent. To our knowledge, this is the first input-based LLM router with distribution-free safety guarantees.

1 Introduction

Large language model routing assigns each query to one of several candidate models, typically balancing response quality against inference cost (Ong et al., 2025; Ding et al., 2024; Chen et al., 2023). Many queries are easy enough for a cheaper model, while only hard queries require an expensive one. A growing body of work develops learned routers that predict which model to use from the input alone (Ong et al., 2025; Chen et al., 2024; Ding et al., 2024).

However, all existing input-based routers answer the same question: how to get the best performance for a given budget. Li et al. (2026) systematically evaluate 10 routing methods across 21 datasets and 33 models, organizing the field into performance-oriented routing and performance-cost trade-off routing. Their key finding is that methods perform similarly under unified evaluation, and none provides formal guarantees on routing quality. We argue that a third axis is missing: what fraction of queries routed to the cheap model will fail? This is the violation rate, and controlling it is critical in deployment settings where a practitioner must certify that the cheap model's failure rate stays below a tolerance α among routed queries.

We formalize this as a safety-constrained routing problem and adapt a proactive conformal gate framework (Uddin et al., 2026) to the LLM setting. A lightweight logistic regression gate, trained on text embeddings, predicts whether routing to the cheap model is safe. A conformal calibration procedure then selects a routing threshold with a distribution-free guarantee:

$$\Pr\big(\Pr(Y{=}0 \mid s(X) \geq t^*) > \alpha\big) \leq \delta, \quad (1)$$

where $Y{=}0$ indicates that routing to the cheap model is unsafe. This guarantee holds regardless of the gate's calibration quality and requires only exchangeability between calibration and test data.

We make three contributions. First, we provide the first input-based LLM router with distribution-free safety guarantees, applying a proactive conformal gate framework to text embeddings and binary correctness labels. Second, we show on two benchmarks that conformal routing controls violations within the δ tolerance across a sweep from 0.05 to 0.50, while a validation-tuned baseline crosses the violation boundary. Third, we present a feasibility analysis across all 10 RouterBench cheap models showing that routability is jointly model- and task-dependent, providing a practical diagnostic for deployment decisions.

Code is available in the supplementary material.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 791–799
July 2-7, 2026 ©2026 Association for Computational Linguistics

Experiments use the publicly available RouterBench dataset (Hu et al., 2024).

Section 2 reviews related work. Section 3 describes the method. Section 4 presents experiments, and Section 5 discusses implications and limitations.

2 Related Work

Input-based LLM routers. Shnitzer et al. (2023) first propose learning a router from benchmark datasets by reducing model selection to binary classification. RouteLLM (Ong et al., 2025) trains matrix factorization and BERT-based routers on human preference data and calibrates a threshold to a budget target. HybridLLM (Ding et al., 2024) predicts query difficulty and tunes a quality threshold on a validation set. RouterDC (Chen et al., 2024) uses dual contrastive learning to align query and model embeddings. LLMRouterBench (Li et al., 2026) benchmarks 10 methods across 21 datasets and finds that methods perform similarly. Their evaluation framework covers performance-oriented routing (maximize quality) and performance-cost routing (best quality per dollar), but does not include guarantee-oriented evaluation (certify bounded failure rate). All these methods optimize cost-performance trade-offs without bounding the failure rate among routed queries.

Cascading and output-based methods. FrugalGPT (Chen et al., 2023) cascades through LLMs, accepting the first sufficiently confident response. This requires running models sequentially and inspecting outputs before deciding whether to escalate. Our method is proactive: the gate examines only the input, avoiding cheap-model inference on escalated queries.

Conformal prediction for LLM routing. Several recent works apply conformal prediction to LLM routing, but differ from ours in key ways. CP-Router (Su et al., 2025) uses the size of conformal prediction sets as an uncertainty proxy to route between a standard LLM and a reasoning model. It requires running the LLM to obtain logits for building prediction sets, making it output-dependent. Its guarantee is marginal coverage ($P(Y \in C(X)) \geq 1 - \alpha$) over all inputs, not a bound on the failure rate among routed queries. RACER (Hao et al., 2026) formulates routing as a risk-controlled set selection problem, constructing calibrated model sets via concentration bounds. RACER also requires base router scores from all candidate models, making it output-dependent, and it controls expected misrouting risk rather than providing a high-probability bound on the violation rate of the routed subset. C3PO (Valkanas et al., 2025) applies conformal prediction to bound the probability that *cost* exceeds a budget in LLM cascades, not that quality stays above a threshold.

Table 1: Positioning of conformal LLM routing methods. "Input" means the router uses only the query text. "Output" means it requires model outputs (logits, predictions, or confidence scores).

Method	Gate	Guarantee	Object
CP-Router	Output	Marginal	Coverage
RACER	Output	Expected	Risk
C3PO	Output	High-prob	Cost
Ours	**Input**	**High-prob**	**Quality**

Table 1 summarizes these distinctions. Our method is the only one that is both input-based (no model outputs needed at routing time) and provides a high-probability guarantee on the violation rate conditional on the routed subset.

Conformal prediction for LLM outputs. A separate line of work uses conformal prediction to quantify uncertainty in LLM outputs rather than for routing. Conformal prediction (Vovk et al., 2005) provides distribution-free uncertainty quantification under exchangeability. Conformal risk control (Angelopoulos et al., 2025) guarantees $\mathbb{E}[L(\lambda)] \leq \alpha$ for monotone losses, controlling expected risk marginally over all test points. Distribution-free risk-controlling prediction sets (Bates et al., 2021) provide high-probability guarantees on prediction sets for a single model. LofreeCP (Su et al., 2024) extends conformal prediction to LLMs without logit access by using sampling-based nonconformity scores. These methods address a different problem: they quantify uncertainty for a single model's outputs rather than controlling quality loss from routing between models. Proactive routing with Clopper-Pearson conformal calibration has been formalized for tabular regression tasks (Uddin et al., 2026). We adapt this framework to LLM routing with binary correctness labels and text embeddings.

3 Method

Problem setup. Let f be an expensive model (e.g., GPT-4) and g a cheap model (e.g., Mixtral-

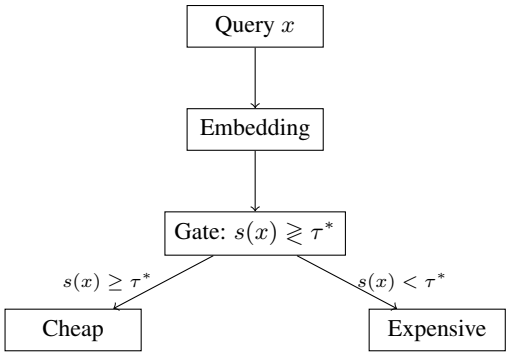

Figure 1: Pipeline overview. Queries above the conformally calibrated threshold τ^* are routed to the cheap model; others go to the expensive model.

8x7B). For input x with ground-truth answer y, define the binary safety label:

$$Y(x) = 1 - \big(\mathbf{1}[g \text{ wrong}] \cdot \mathbf{1}[f \text{ correct}]\big). \quad (2)$$

That is, $Y = 0$ (unsafe to route to cheap) only when the cheap model is wrong *and* the expensive model is correct. If both models are wrong, routing to the cheap model incurs no additional loss compared to the expensive model, so $Y = 1$. If the cheap model is correct, $Y = 1$ regardless. This definition captures the intuition that routing to the cheap model only hurts when the expensive model would have done better.

Let $\pi = \Pr(Y{=}1)$ be the base safe rate, the fraction of queries where routing to the cheap model is safe. A routing policy $\hat{\pi}(x) = \mathbf{1}[s(x) \geq t]$ routes to the cheap model when gate score $s(x)$ exceeds threshold t. The violation rate among routed inputs is:

$$V(t) = \Pr(Y{=}0 \mid s(X) \geq t). \quad (3)$$

See Figure 1 for an overview of the routing pipeline.

Gate training. We embed each query using BAAI/bge-base-en-v1.5 (Xiao et al., 2023), a general-purpose text embedding model (512 max tokens, 768 dimensions), and train a logistic regression to predict Y from the embedding. The gate outputs $s(x) = \hat{p}(Y{=}1 \mid x)$.

Conformal threshold selection. We use a held-out calibration set $\{(s_i, Y_i)\}_{i=1}^n$ to select the lowest threshold t^* where the Clopper-Pearson upper confidence bound on the violation rate is at most α:

$$t^* = \min\big\{t \in \{s_1, \ldots, s_n\} : \text{UCB}_\delta(k(t), n(t)) \leq \alpha\big\}, \quad (4)$$

where $n(t) = |\{i : s_i \geq t\}|$, $k(t) = |\{i : s_i \geq t, Y_i = 0\}|$, and $\text{UCB}_\delta(k, n) = B^{-1}(1{-}\delta;\, k{+}1,\, n{-}k)$ is the exact Clopper-Pearson upper bound. If no valid t exists, $t^* = \infty$ and nothing is routed. The guarantee $\Pr(V(t^*) > \alpha) \leq \delta$ follows from the Clopper-Pearson coverage property. We use an ascending threshold search and verify empirically that $V(t)$ is monotone non-increasing up to the selected threshold t^* on both datasets (99.3% of steps on GSM8K, 99.8% on MMLU), consistent with the conformal risk control monotonicity condition (Angelopoulos et al., 2025). A stricter descending fixed-sequence approach under Learn-Then-Test (Angelopoulos et al., 2022) that does not require this condition is discussed in Uddin et al. (2026).

Feasibility condition. The constraint $V(t) \leq \alpha$ is satisfiable with positive coverage if and only if there exists t such that $\text{TPR}(t)/\text{FPR}(t) \geq C(\pi, \alpha)$, where:

$$C(\pi, \alpha) = \frac{(1-\pi)(1-\alpha)}{\pi\alpha}. \quad (5)$$

When π is high (most queries are safe), C is small and weak gates suffice. When π is low, only gates with strong separation can route safely. This provides a practical diagnostic: compute C from the data before investing in gate training (Uddin et al., 2026).

4 Experiments

4.1 Setup

We use RouterBench (Hu et al., 2024), which provides pre-computed correctness labels for 11 models across 8 datasets. We route between Mixtral-8x7B-Instruct (cheap, mean cost \$0.0013/query) and GPT-4-1106-preview (expensive, mean cost \$0.0319/query), a 24.5× cost difference based on mean per-query token costs from RouterBench. We evaluate on two datasets: GSM8K (Cobbe et al., 2021) (7,450 queries, grade-school math) and MMLU (Hendrycks et al., 2021) (14,042 queries, 57 sub-topics combined).

The safety label follows Eq. 2: scores ≥ 0.5 are binarized as correct, and $Y = 0$ only when Mixtral is wrong and GPT-4 is correct. We set per-dataset

Table 2: Test-set results. Coverage is the fraction routed to the cheap model. Violation is the failure rate among routed queries (Eq. 3). ✓ means $V \leq \alpha$. ✗ means violated. Savings are relative to always-expensive.

Dataset	Method	Cov.	Viol.	$\leq\alpha$?	Save
GSM8K (α=0.30)	Conformal	.367	.280	✓	35%
	Val-Tuned	.596	.317	✗	58%
	Naive	1.00	.354	✗	97%
	Oracle	.646	.000	✓	63%
MMLU (α=0.20)	Conformal	.903	.191	✓	87%
	Val-Tuned	.926	.196	✓	90%
	Naive	.993	.215	✗	96%
	Oracle	.784	.000	✓	76%

tolerances: $\alpha = 0.30$ for GSM8K (harder task, more tolerance needed) and $\alpha = 0.20$ for MMLU, with $\delta = 0.10$ throughout.

Queries are split into four parts (55/15/15/15%) for training, calibration, validation, and testing with stratified sampling on Y. The gate is logistic regression with default scikit-learn parameters (L2 penalty, $C = 1.0$). The four-way split ensures no data leaks between gate training, conformal calibration, val-tuned threshold selection, and final evaluation.

4.2 Baselines

Conformal (ours): Clopper-Pearson threshold selection on the calibration set. *Val-Tuned*: lowest threshold on the validation set achieving $V \leq \alpha$, representing the standard practice in existing routers. *Naive* (t=0.5): route if $s(x) \geq 0.5$. *Oracle*: route if and only if Y=1. *Always-Cheap / Always-Expensive*: route all or none.

4.3 Routing Safety and Coverage

Table 2 shows the main results. On GSM8K, the conformal method satisfies $V \leq \alpha = 0.30$ with 36.7% coverage, while the val-tuned baseline exceeds α at 0.317 despite achieving higher coverage. On MMLU, both conformal and val-tuned satisfy $\alpha = 0.20$, but only conformal provides the formal guarantee. Cost savings reach 87% on MMLU, where conformal routes 90% of queries to Mixtral at 24.5× lower per-query cost.

The GSM8K result is the key finding. Val-tuned routing, which is the standard practice, violates the target in deployment. The conformal method trades coverage for safety, which is the correct behavior when guarantees matter.

Table 2 also shows the cost implications.

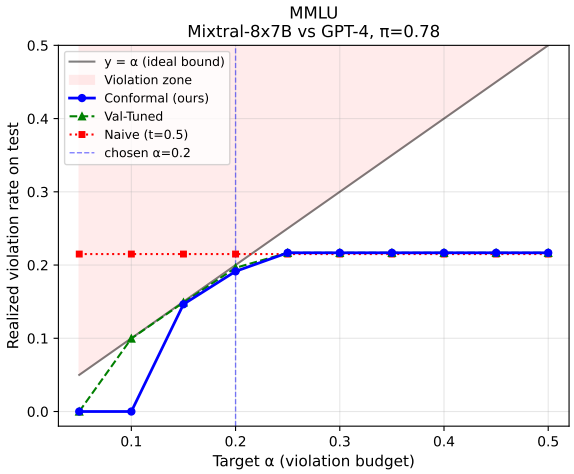

Figure 2: MMLU: test violation vs. target α (0.05 to 0.50). Conformal (blue) stays near or below the diagonal, consistent with the $\delta = 0.10$ tolerance. Val-tuned (green) stays close but has no formal guarantee. Results are from a single stratified split.

Always-expensive routing costs $0.0319 per query with zero violations. Our conformal approach reduces cost by 87% on MMLU (violation 0.191) and 35% on GSM8K (violation 0.280), both within the α tolerance. Always-cheap routing saves 97% but violates on both datasets.

4.4 α-Sweep Analysis

Figures 2 and 3 show the central result. Across $\alpha \in [0.05, 0.50]$, the conformal method's test violation stays near or below α on both datasets, consistent with the $\delta = 0.10$ tolerance. The val-tuned baseline crosses the diagonal on GSM8K around $\alpha = 0.20$ and $\alpha = 0.30$. This is consistent with the nominal $\delta = 0.10$: the conformal guarantee holds, while empirical threshold tuning provides no such assurance.

On MMLU (Figure 2), both methods stay below the diagonal because the task is relatively easy ($\pi = 0.784$) and the gate has decent discriminative power (AUC = 0.678). On GSM8K (Figure 3), the harder task ($\pi = 0.646$) and weaker gate (AUC = 0.578) create tighter conditions where val-tuned's lack of statistical margin causes it to fail.

4.5 Coverage-Violation Trade-off

Figures 4 and 5 show the coverage-violation trade-off. On MMLU, the conformal method routes 90% of queries to the cheap model while staying safely below $\alpha = 0.20$. The gap between conformal and val-tuned is small (0.90 vs. 0.93). On GSM8K, conformal achieves lower coverage

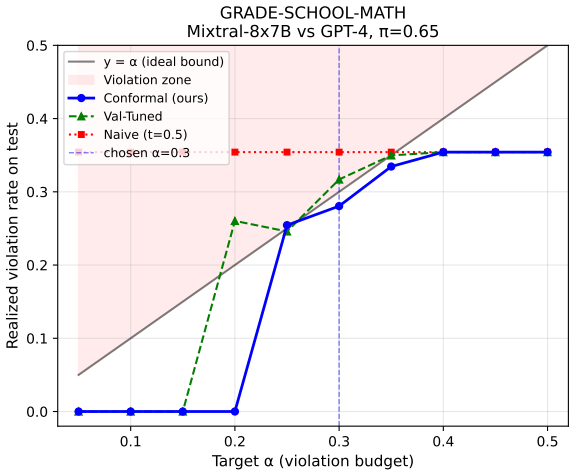

Figure 3: GSM8K: test violation vs. target α. Conformal (blue) stays near or below the diagonal, consistent with the $\delta = 0.10$ tolerance. Val-tuned (green) crosses the diagonal around $\alpha = 0.20$ and $\alpha = 0.30$. Results are from a single stratified split.

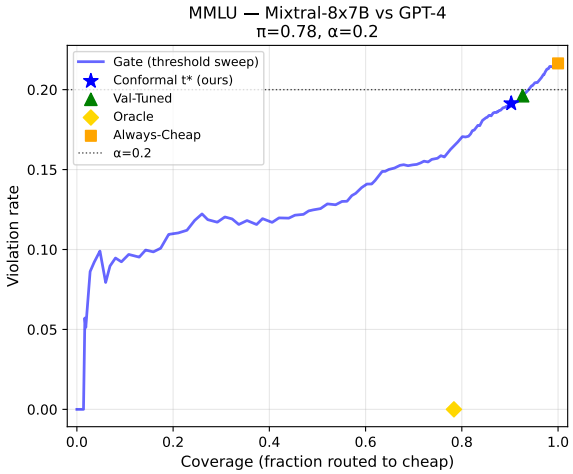

Figure 4: MMLU: coverage vs. violation for each method at $\alpha = 0.20$. Conformal operates below the α line with 90% coverage.

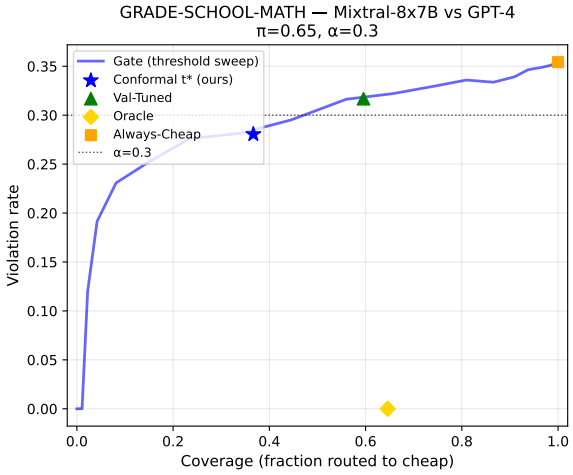

Figure 5: GSM8K: coverage vs. violation at $\alpha = 0.30$. Conformal operates safely at 37% coverage. Val-tuned achieves 60% coverage but violates.

(0.37 vs. 0.60 for val-tuned), reflecting the price of the guarantee on a harder task where the gate has less discriminative power.

4.6 Gate ROC Analysis

Figures 6 and 7 show the gate ROC curves with the feasibility slope $C(\pi, \alpha)$ overlaid. On MMLU, the gate AUC (0.678) comfortably exceeds the sufficient threshold $\Phi_c^* = 0.551$, guaranteeing feasibility under any concave ROC assumption. On GSM8K, the gate AUC (0.578) falls below $\Phi_c^* = 0.608$. Routing still succeeds because feasibility depends on the local ROC slope at the operating point, not on global AUC. The initial portion of the GSM8K ROC curve is steep enough to exceed $C = 1.28$, enabling a positive coverage threshold. This illustrates that the sufficient AUC condition is conservative, and gates that fail the global test may still route successfully.

4.7 Feasibility Analysis

Table 3 shows the critical ratio $C(\pi, \alpha)$ for all 10 cheap models in RouterBench. This diagnostic is useful even without running the full conformal pipeline: if $C \gg 1$, safe routing likely requires a very strong gate or is infeasible entirely.

For example, code-llama-34b has $C = 10.92$ on MMLU because it answers only 26.8% of MMLU queries correctly, making safe routing essentially impossible. Conversely, claude-v2 on GSM8K has $C = 0.17$ because it answers 93.2% correctly, so almost any gate suffices, but the routing savings are minimal because the cheap model rarely needs help.

The table demonstrates that feasibility is jointly model- and task-dependent. Code-llama-34b has $C = 2.75$ on GSM8K but $C = 10.92$ on MMLU, while mixtral-8x7b shows the opposite pattern ($C = 1.28$ vs. 1.10). This joint dependency means that feasibility cannot be assessed from model quality alone. It requires task-specific analysis with the C diagnostic.

5 Discussion

The guarantee is the contribution. Conformal routing achieves lower coverage than val-tuned on GSM8K (0.37 vs. 0.60). This is expected and cor-

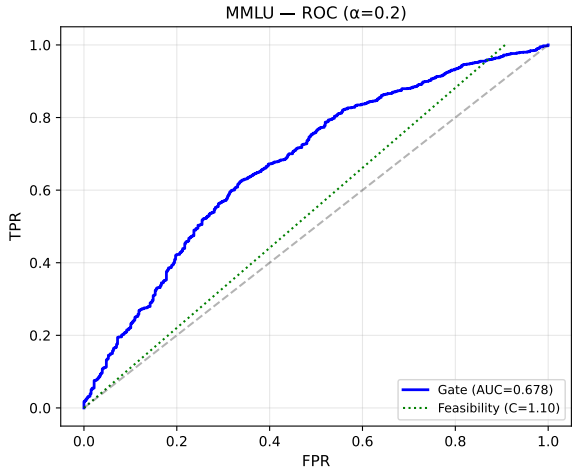

Figure 6: MMLU gate ROC curve (AUC = 0.678). The dashed line shows the feasibility slope $C = 1.10$. The ROC curve exceeds this slope initially, confirming feasibility.

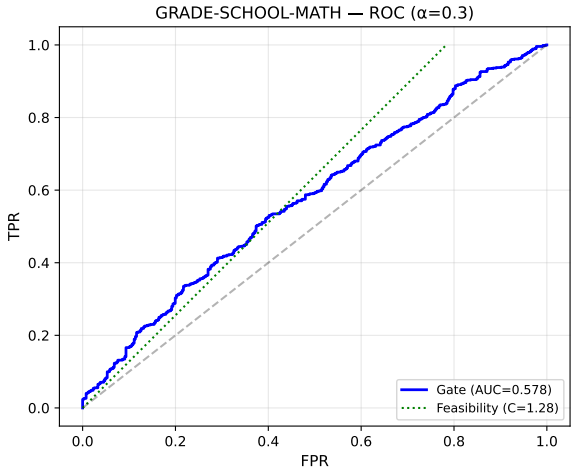

Figure 7: GSM8K gate ROC curve (AUC = 0.578). The feasibility slope $C = 1.28$ is steeper. The global AUC falls below the sufficient threshold $\Phi_c^* = 0.608$, yet routing succeeds because the initial ROC slope locally exceeds C.

Table 3: Feasibility diagnostic: $C(\pi, \alpha)$ for all 10 RouterBench cheap models vs. GPT-4. Lower C means easier to route safely. $C > 2$ is typically infeasible with a logistic gate.

Cheap Model	GSM8K ($\alpha = 0.30$)		MMLU ($\alpha = 0.20$)	
	π	C	π	C
claude-v2	.932	0.17	.716	1.59
gpt-3.5	.736	0.84	.752	1.32
mixtral-8x7b	.646	1.28	.784	1.10
llama-2-70b	.547	1.93	.606	2.60
mistral-7b	.487	2.46	.657	2.09
code-llama-34b	.459	2.75	.268	10.92
zephyr-7b	.445	2.91	.581	2.88
phi-2	.419	3.24	.474	4.44
llama-2-13b	.314	5.10	.523	3.65
llama-2-7b	.217	8.44	.406	5.86

rect: the guarantee comes at the cost of coverage. When coverage is the only metric, val-tuned wins. When a practitioner needs to certify that the failure rate stays below α, conformal is the only option among the methods we test.

Connection to LLMRouterBench evaluation axes. Li et al. (2026) organize LLM routing evaluation into performance-oriented routing and performance-cost trade-off routing. Our binary safety label Y (Eq. 2) connects naturally to performance: the violation rate $V(t)$ measures the rate at which routing to the cheap model causes a correctness loss that the expensive model would have avoided. Controlling this rate introduces a third evaluation axis, guarantee-oriented routing, where the goal is to certify that quality degradation stays within bounds rather than to maximize performance or minimize cost.

Comparison with conformal routing methods. CP-Router (Su et al., 2025) and RACER (Hao et al., 2026) both apply conformal prediction to LLM routing but operate in a fundamentally different regime. Both are output-dependent: CP-Router requires LLM logits to construct prediction sets, and RACER requires base router scores from all candidate models. Our gate operates on the input alone, making a routing decision before any model is called. This distinction matters in practice because output-dependent methods must run the cheap model (or all candidates) on every query, negating the cost savings from routing when the query is escalated. Furthermore, their guarantees differ from ours: CP-Router provides marginal coverage over all inputs, RACER controls expected risk, and ours provides a high-probability bound on the violation rate of the routed subset (Eq. 1).

Future work. Multi-model routing DAGs with per-branch conformal guarantees could extend this framework beyond two-model routing. Richer safety labels such as graded quality scores would allow finer control over what counts as a violation, and would address open-ended generation settings where correctness is not binary. Distribution shift at deployment time is an important limitation of the exchangeability assumption, and adaptive conformal inference (Gibbs and Candès, 2021) provides a principled extension maintaining cover-

age under temporal shift. Conditional conformal methods could provide subgroup-level guarantees rather than marginal ones. Combining input-based gates with output-dependent verification could improve coverage while preserving guarantees. Ablation studies on calibration set size, embedding model choice, and gate complexity remain for future work.

Limitations

The guarantee provided by conformal calibration is marginal over calibration randomness, not conditional on input subregions. This means that certain subgroups of queries (e.g., queries from a specific MMLU sub-topic) may have higher violation rates than the global α, even though the aggregate guarantee holds. Binary correctness is a coarse safety label that does not capture partial quality differences between models. For example, a response that is mostly correct but contains a minor error is treated the same as a completely wrong response. The exchangeability assumption required by conformal prediction may not hold under distribution shift at deployment time, for instance if the query distribution changes after calibration. Gate quality directly limits coverage: our logistic gate achieves AUC 0.578 on GSM8K, and a stronger gate (e.g., a nonlinear model trained on a better embedding) could improve coverage without affecting the guarantee. We evaluate on only two datasets and one model pair from RouterBench, so the generality of our findings to other tasks and model combinations remains to be verified. Finally, our method requires pre-computed correctness labels for both models on a calibration set, which may not always be available in practice. Results are reported on a single stratified split and stability across multiple splits remains to be verified given the probabilistic nature of δ. The ascending threshold search relies on an empirically verified monotonicity condition (99.3% of steps on GSM8K, 99.8% on MMLU).

References

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. 2022. Learn then test: Calibrating predictive algorithms to achieve risk control. *Preprint*, arXiv:2110.01052.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. 2025. Conformal risk control. *arXiv preprint arXiv:2208.02814*.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. 2021. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34.

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

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. 2024. RouterDC: Query-based router by dual contrastive learning for assembling large language models. *arXiv preprint arXiv:2409.19886*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed H Awadallah. 2024. Hybrid LLM: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Isaac Gibbs and Emmanuel Candès. 2021. Adaptive conformal inference under distribution shift. *Preprint*, arXiv:2106.00170.

Sai Hao, Hao Zeng, Hongxin Wei, and Bingyi Jing. 2026. RACER: Risk-aware calibrated efficient routing for large language models. *arXiv preprint arXiv:2603.06616*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, Zhen Wang, and Shuyue Hu. 2026. LLMRouterBench: A massive benchmark and unified framework for LLM routing. *arXiv preprint arXiv:2601.07206*.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs with preference data. *arXiv preprint arXiv:2406.18665*.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*.

Jiayuan Su, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. 2025. CP-Router: An uncertainty-aware router between LLM and LRM. *arXiv preprint arXiv:2505.19970*.

Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. 2024. API is enough: Conformal prediction for large language models without logit-access. In *Findings of EMNLP*.

Iqtedar Uddin, Mazin Khider, and André Bauer. 2026. Proactive routing to interpretable surrogates with distribution-free safety guarantees. *arXiv preprint arXiv:2603.14623*.

Antonios Valkanas, Soumyasundar Pal, Pavel Rumiantsev, Yingxue Zhang, and Mark Coates. 2025. C3po: Optimized large language model cascades with probabilistic cost constraints for reasoning. *arXiv preprint arXiv:2511.07396*.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. 2005. *Algorithmic Learning in a Random World*. Springer.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

A Safety Label Details

The safety label is computed from RouterBench's pre-computed scores. Scores ≥ 0.5 are binarized as correct. The label follows Eq. 2 and is implemented as:

$$Y = 1 - ((1 - \text{cheap\_correct}) \times \text{exp\_correct}), \quad (6)$$

so $Y = 0$ only when the cheap model is wrong and the expensive model is correct. Table 4 shows the breakdown of cases for both datasets.

Table 4: Safety label breakdown. The four cases correspond to joint correctness of the cheap (g) and expensive (f) models.

g	f	Y	Interpretation
Correct	Correct	1	Safe (cheap suffices)
Correct	Wrong	1	Safe (cheap is better)
Wrong	Wrong	1	Safe (no loss from routing)
Wrong	Correct	0	**Unsafe** (cheap loses)