PREDICTABLE BUT NOT ACTIONABLE: A DECISION-GAP ANALYSIS OF STATIC LLM ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) routing routes queries to balance quality and cost, yet static routers plateau below achievable quality. Why this plateau persists, and how routing should be evaluated, remains open. Existing evaluations are flawed because they score routers as predictors, not policies whose switches determine value. A corrected standard separates genuine gains from measurement artifacts before deployment. We formalize the decision gap $D(q)$, the oracle improvement over a stated prior, and decompose value into switch rate and per-switch gain. On RouterBench (36,497 queries), we evaluate query-only readers, prefill probes, and a two-stage cascade. Our protocol separates oracle-aware from selection-valid evaluation via deployable inputs. Shuffle controls and nested cross-validation govern comparisons. Ten-seed sweeps, calibration audits, and leakage-free replay validate the findings. We find headroom on 18.1% of queries, readers flipping a third of decisions for near-zero gain, and one cascade gaining under oracle-aware but losing under leakage-free replay.

1 INTRODUCTION

The ecosystem of large language models spans an order of magnitude in cost, which has made *routing*, the choice of which model answers each query, a central mechanism for balancing quality against cost Hu et al. (2024); Ong et al. (2025). The dominant design is a learned reader that maps the query to a model choice, trained on per-model outcomes and evaluated by prediction accuracy or attained quality. Yet across reader families, architectures, and supervision targets, static query-side routers converge to a narrow band of quality Hu et al. (2024); Lu et al. (2026b;a). What has been missing is an explanation of *why* the band exists and *where* the missing value is located.

This paper argues that the plateau is a measurement problem before it is a modeling problem. Routing is a decision problem: a router creates value only when it selects a *different* model whose answer is *better*. Prediction metrics do nottest this: a reader can rank models accurately on average while making exactly the switches that are useless or harmful. The relevant target is the improvement available over a stated deployment policy, and that improvement is concentrated on a sparse subset of queries.

We formalize this target as the *decision gap*

$$D(q) \;=\; \max_{m \in M} s_m(q) \;-\; s_{m_0}(q), \tag{1}$$

where $s_m(q)$ is the benchmark score of candidate m on query q and m_0 is the *prior*, the model a deployment would use without a router, which we take to be the strongest model on the training fold. $D(q)$ is an oracle diagnostic: it tells us *where* switching can help, not *how* to decide to switch. It partitions the benchmark into a small decision-bearing subset and a large region where the prior is already row-optimal. Figure 1 summarizes the resulting argument.

Working on the released RouterBench score matrices Hu et al. (2024), we obtain four findings. **First**, the oracle headroom is large in aggregate ($+0.131$ mean quality) but sparse in decisions: only 18.1% of 0-shot queries have $D(q) > 0$. **Second**, query-only readers predict residual variation well ($R^2 \approx 0.134$) yet their induced switches are nearly break-even—flipping 34.0% of decisions yields a mean gain of about 0.8% of the headroom—and re-targeting the reader to predict $D(q)$ raises R^2 to 0.178 but leaves routed quality fixed at the prior band. **Third**, the target and representation matter

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

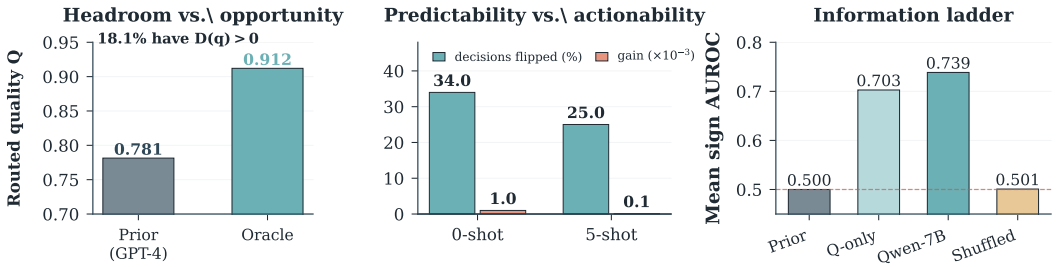


Figure 1: **The decision-gap view of LLM routing. Left:** on RouterBench 0-shot the prior (GPT-4) attains Q=0.781 and the oracle Q=0.912, but only 18.1% of queries admit a strictly better candidate. **Middle:** across ten splits, query-only readers flip $34.0\%/25.0\%$ of decisions (0/5-shot) while the mean routed gain stays within ± 0.002 of zero. **Right:** mean gap-detection AUROC across representations: 7B hidden states lift row-level gap detection from 0.539 to 0.567; shuffling collapses it to 0.493.

more than the reader: continuous targets recover the prior band where hard labels collapse on the same architecture, and hidden states of a 7B LLM raise per-model score discrimination substantially, with a row-shuffle control collapsing to chance. **Fourth**, the diagnostic/deployable boundary is sharp: a cascade whose trigger uses an oracle quantity beats the prior on a diagnostic subset, but a strictly leakage-free replay, whose trigger is restricted to the query embedding at test time, ends *below* the prior on the full pool.

These findings jointly reframe what routing evaluations must report: headroom relative to a stated prior, predictability under a named information set, and selection-valid utility of the induced decisions. A router claim is complete only when all three are stated.

Contributions.

- **Decision-gap framework.** We define the decision gap, its relation to routed quality through a switch decomposition (Section 2), and a protocol that separates oracle-aware diagnostics from selection-valid evaluation. The framework applies to any score-matrix benchmark and any stated prior.
- **Empirical diagnosis of the routing plateau.** On RouterBench we show that (i) decision opportunities are sparse (18.1%) and prior-relative ($18\% \rightarrow 69\%$ as the prior weakens); (ii) query-only readers flip a third of decisions for near-zero gain, persisting under direct decision-gap targets; (iii) hard labels discard within-row structure that continuous targets preserve; (iv) representation quality, not reader capacity, governs predictability (Section 4).
- **Information-boundary evidence.** Candidate-conditioned hidden states carry query–model interaction signal that frozen query encoders do not (score-discrimination AUROC $0.703 \rightarrow 0.739$, shuffle control 0.501), and a leakage-free replay shows this signal does not yet convert into full-pool selection-valid gain (Sections 5–6).

Section 2 develops the framework and protocol. Section 4 presents the query-only diagnosis. Section 5 locates stronger signal in candidate-conditioned representations. Section 6 separates diagnostic headroom from selection-valid routing. Section 3 positions the work among recent routing research, and Section 7 discusses implications, limitations, and the evaluation standard that follows from the results.

2 The Decision-Gap Framework

2.1 Setting and quantities

Let $M = \{1, \ldots, |M|\}$ be a pool of candidate models and let $s_m(q) \in [0, 1]$ denote the benchmark score of model m on query q. In a score-matrix benchmark such as RouterBench Hu et al. (2024), the matrix $\{s_m(q)\}$ is released with one column per model. A *router* is a policy f that maps an admissible test-time representation of q to a model choice $\hat{m}(q) \in M$. Its *routed quality* is

$$Q_f \;=\; \mathbb{E}_q\big[s_{\hat{m}(q)}(q)\big]. \tag{2}$$

The *prior* m_0 is the policy a deployment would run without a router; we take $m_0 = \arg\max_m \mathbb{E}_{q\in\text{train}}[s_m(q)]$, the strongest model on the training fold, and throughout RouterBench this is GPT-4. The prior quality is $Q_{\text{prior}} = Q_{m_0}$ and the *oracle* is the per-query best selection $m^\star(q) = \arg\max_m s_m(q)$ with quality Q_{oracle}. The *decision gap* of query q, defined in Eq. equation 1, satisfies two identities that make it the right bridge between prediction and decisions:

$$Q_{\text{oracle}} - Q_{\text{prior}} \;=\; \mathbb{E}_q\big[D(q)\big], \qquad \Pr\big[D(q) > 0\big] \;=\; \Pr\big[m_0 \neq m^\star(q)\big]. \tag{3}$$

The first identity decomposes aggregate headroom into a per-query quantity; the second shows that $D(q)$, not the oracle label, identifies where a router can act at all. We call a query *decision-bearing* when $D(q) > 0$.

2.2 Actionability: the switch decomposition

The central distinction of this paper is between *predictability* (how well a reader forecasts scores or gaps) and *actionability* (whether its induced switches improve quality). Any reader f admits the exact decomposition

$$Q_f - Q_{\text{prior}} \;=\; \underbrace{\Pr\big[\hat{m}(q) \neq m_0\big]}_{\text{switch rate}} \cdot \mathbb{E}\big[\, s_{\hat{m}(q)}(q) - s_{m_0}(q) \;\big|\; \hat{m}(q) \neq m_0 \,\big]. \tag{4}$$

The expectation factorizes into *beneficial* switches ($s_{\hat{m}} > s_{m_0}$, possible only when $D(q) > 0$) and *harmful* switches ($s_{\hat{m}} < s_{m_0}$, possible even when $D(q) = 0$, where any switch is a pure loss). Eq. equation 4 explains how a reader can be predictive and still useless: a high switch rate multiplied by a near-zero (or negative) conditional switch gain produces no routed improvement. Our experiments report both factors, so the plateau is not an aggregate accident but a measurable mismatch between when readers switch and when switching pays. Prediction metrics measure a third, intermediate quantity: R^2 on scores, and the *gap-detection AUROC*, the row-level ranking quality for $\mathbf{1}[D(q) > 0]$, distinct from the *score-discrimination AUROC* used for per-(query, model) correctness below.

2.3 Information sets and protocol validity

The value of a router depends on what it is allowed to see, so every quantity above is relative to an *information set*. We distinguish four: Four information sets organize the paper: $\mathcal{I}_{\text{query}}$ (features of the query alone: text embeddings, frozen-encoder or LLM-encoder activations; available before any model is invoked); $\mathcal{I}_{\text{cand}}$ (per-candidate *prefill* states $h_m(q)$, one forward pass of candidate m on q before generation; strictly stronger, but costly and unavailable for closed APIs); $\mathcal{I}_{\text{resp}}$ (the prior's observed response or score on the same query; available only after the prior answers); and $\mathcal{I}_{\text{oracle}}$ (the full observed score-matrix row; upper bounds only). A reader trained on labels derived from outcomes (as all readers are) is deployable only if, on a held-out query, its inputs are drawn from the declared information set. We call an evaluation *oracle-aware* if any input is drawn from $\mathcal{I}_{\text{resp}}$ or $\mathcal{I}_{\text{oracle}}$ for the evaluated query, and *selection-valid* (leakage-free) otherwise. Both are legitimate measurements; conflating them is the error our protocol is designed to prevent. All cascade-style numbers in this paper carry an explicit information-set label.

2.4 Benchmark, splits, and readers

We use RouterBench 0-shot (36,497 queries, 11 models) as the primary matrix and 5-shot (36,511) as the replication split; candidates span WizardLM-13B through GPT-4 (Appendix A). Query features are MiniLM-L6 embeddings (384-d) unless stated otherwise. Reader comparisons use stratified five-fold cross-validation (seed 7); the multi-seed analysis uses ten independent 80/20 splits (seeds 0–9). All preprocessing, target construction, scaling, and threshold selection are fit inside training folds. The reader catalogue spanssix families (retrieval aggregation, hard-label classifiers, continuous-target classifiers, multi-output regression, pairwise ranking, and IRT) with all hyperparameters fixed except k (inner validation); the full grid is in Appendix B.

Two properties of the released matrix shape all conclusions. First, the judge is coarse: 93.0% of 0-shot rows are tied at the row maximum, so hard labels carry little within-row signal; a 200-query continuous rejudging audit drops the tied-best rate to 20% (Appendix D), making tie-structure

findings judge-resolution-specific. Second, the pool is fixed ($|M| = 11$), so prior-relative statements are exactly that: the opportunity set changes with the prior (Section 4.5). We make no claims about live traffic, latency-sensitive settings, or open pools (Section 7).

2.5 Evaluation regions

For descriptive analysis we partition rows with pass threshold $t = 0.5$ and tier maxima S_{small}, S_{strong} into *trivial*, *routable*, and *strong-failure* ($S_{\text{small}} \geq t$; $S_{\text{small}} < t \leq S_{\text{strong}}$; $S_{\text{strong}} < t$). Regions are computed from observed outcomes and used only to *locate* phenomena, never as router inputs.

3 Related Work

Routing benchmarks and the plateau. RouterBench Hu et al. (2024) introduced the score-matrix evaluation we adopt and documented that published routers recover a small fraction of the oracle gap; the Routing Plateau Lu et al. (2026b) showed that simple readers match learned ones; RouterArena Lu et al. (2026a) reports that higher router accuracy does not imply a better accuracy–cost rank. Our results give the plateau a mechanism: the decision-bearing subset is sparse and prior-relative, and query-only signal does notidentify it, reframing "better routers" as a problem of information and selection validity rather than reader architecture.

Query-side routers. RouteLLM Ong et al. (2025) learns strong/weak routers from preference data; Universal Model Routing Jitkrittum et al. (2026) routes to unseen models with excess-risk guarantees; kNN baselines remain competitive Li (2025). These methods optimize prediction of model quality; the decision gap measures what predictions are worth as *switches* against a stated prior, and the two can diverge at high prediction quality. The contrast with unseen-model routing is complementary: prior-relativity (Section 4.5) implies that dynamic pools change the opportunity set with every portfolio, so headroom statistics should be reported per portfolio, not only per benchmark. Reporting switch rate and gain-per-switch (Eq. equation 4) alongside prediction metrics would sharpen such comparisons.

Model-aware, ability-aware, and interaction-aware routing. DiSRouter Zheng et al. (2026) distributes routing decisions across models; GraphPlanner Feng et al. (2026) adds graph memory for agentic routing; RADAR Fernandez et al. (2026) conditions selection on reasoning ability and difficulty; CITER Zheng et al. (2025) routes during decoding. These systems exploit information beyond the query, and our probe results (Section 5) give that design an information-theoretic footing: candidate-conditioned states carry interaction signal that query-only features lack ($0.703 \rightarrow 0.739$ AUROC, shuffle control 0.501). The decision gap provides the audit those works can adopt: how much oracle headroom is realizable under the declared test-time signal?

Cascades and selective escalation. FrugalGPT Chen et al. (2024) popularized LLM cascades; the cascade-routing formulation of Dekoninck et al. Dekoninck et al. (2025) derives optimal policies from quality estimators and identifies estimator quality as the decisive factor; Routing, Cascades, and User Choice Mahmood (2026) analyzes provider-level economics; UCCI Kotte (2026) shows that calibration-first escalation can be cost-optimal under explicit assumptions; conformal schemes add guarantees Dou et al. (2026); Pona et al. (2026). Our results give these frameworks a measurement footing: the estimator quality they depend on is exactly the gap-detection quality of Section 4.3 (AUROC 0.68), and under an 18% base rate the break-even precision of 0.68 (Appendix F) explains why an apparently strong offline trigger loses selection-validly (Table 3). Our switch decomposition and information-set protocol are a concrete instantiation of this audit principle for score-matrix benchmarks.

Positioning. Prior work establishes the plateau, broadens routing mechanisms, and sharpens cascade economics; none reports the sparse-decision diagnosis together with the oracle-aware/selection-valid contrast on the same benchmark. The decision gap makes the three comparable: what benchmarks measure implicitly, what routers estimate, and what cascades act on.

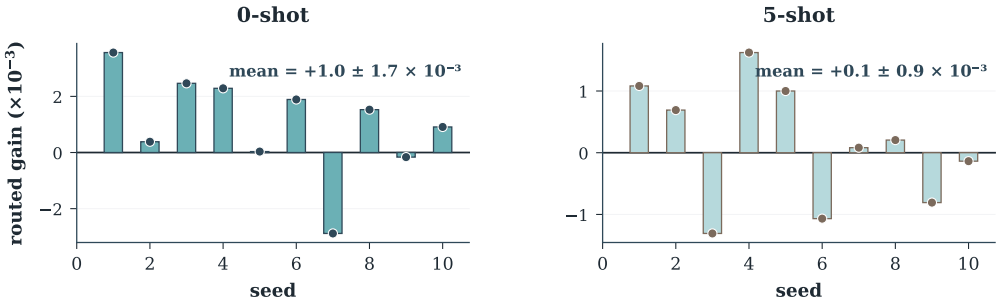


Figure 2: Ten-seed routed gain of the residual reader over the prior. Bars are individual splits; the mean (upper right) is ± 1 seed-level standard deviation. Gains straddle zero on both splits.

4 Query-Only Readers Predict but Do Not Improve Decisions

4.1 The prior absorbs the shared performance trend

Table 1 places reference policies and the strongest query-only readers on one surface. Every continuous-target query-only reader lands within ± 0.003 of the prior $Q_{\text{prior}} = 0.7814$ (oracle 0.9121): Ridge on correctness reaches 0.7819 and retrieval aggregation 0.7840—a *prior band* Lu et al. (2026b) that spans the reader catalogue. This is the first diagnostic result: readers recover the population-level ordering of models (which is what makes them look good under score-prediction metrics) but extract no query-specific correction that improves the selected score.

Table 1: Canonical RouterBench 0-shot results (full pool). Hard-label readers collapse because a single argmax label discards within-row ordering (Section 4.4).

Policy	Target	Q	ΔQ vs. prior	Switch rate	Gain per switch
Reference policies					
Global prior (GPT-4)	—	0.7814	0	0	—
Oracle (row max)	—	0.9121	+0.1307	0.181	+0.7209
Query-only readers (continuous targets, 10-seed mean±std)					
Ridge residual	correctness	0.7814	$+0.0010 \pm 0.0017$	0.340	+0.0029
Query-only readers (hard best-model label)					
KNN-20	argmax	0.519	−0.262	—	—
LogReg-PCA64	argmax	0.510	−0.271	—	—
MLP-CE	argmax	0.522	−0.259	—	—

4.2 Decision-bearing headroom is sparse

The aggregate gap of 0.1307 concentrates on 18.1% of queries; conditional on $D(q) > 0$, the mean available improvement is 0.721: the oracle is much better than the prior exactly where it is better at all. Sparsity has two consequences: a router's maximum gain per switch is large, so the plateau cannot be explained by small per-switch prizes; the failure must lie in *selection*; and score-prediction metrics are dominated by the many rows where the prior is already optimal, so a reader can score highly while contributing nothing, or negatively, on the rows that matter.

4.3 Predictive readers switch often and gain nothing

For each of ten 80/20 splits, a Ridge reader trained on per-model correctness (residualized against training-fold means) is compared with the prior by its $\arg\max$ policy (Figure 2). On 0-shot it flips $33.99\% \pm 0.66\%$ of decisions but gains only $+0.0010 \pm 0.0017$; the 5-shot replication flips 25.02% and gains $+0.0001 \pm 0.0009$ (Figure 2), with per-seed gains straddling zero: the gain's sign is unstable. By Eq. equation 4, the conditional gain per switch is $+0.0029$, indistinguishable from break-even, so the switches are essentially uninformed wagers on a set where 81.9% of wagers are guaranteed losses.

Re-targeting to the gap raises R^2 but not decisions. A natural objection is that the reader is trained on the wrong target. Predicting the decision gap $D(q) = s_m(q) - s_{m_0}(q)$ directly raises R^2

Table 2: Direct prediction of decision-changing targets (0-shot, 10-seed mean±std). Raising R^2 on the gap does not move routed quality: the routing decision depends on the gap's sign, not its magnitude.

Training target	R^2	Routed Q	Gain vs. prior
Raw correctness s_m	0.134 ± 0.005	0.7814	$+0.0010 \pm 0.0017$
Residual $s_m - \mu_m$	0.134 ± 0.005	0.7814	$+0.0010 \pm 0.0017$
Decision gap $s_m - s_{m_0}$	$\mathbf{0.178 \pm 0.003}$	0.7814	$+0.0010 \pm 0.0017$

substantially, yet routed quality and gain stay exactly at the raw-target reader's values (Table 2). The reason is sign selection: routing needs $\mathbf{1}[\hat{s}_m > \hat{s}_{m_0}]$ to be right on decision-bearing rows, and the gap-detection AUROC on $\mathbf{1}[D(q) > 0]$ is only 0.539 (MiniLM) to 0.567 (best tested representation), far from what an 18% base rate requires (Section 5). Predicting the gap's *size* is easy; predicting *which* rows those are is not achieved by any tested query-only reader.

4.4 Hard labels discard the structure continuous targets preserve

The prior band has a sharp boundary at the target. On the same MLP architecture, a tie-broken best-model label yields $Q \approx 0.51$–0.54 while a softmax-normalized correctness vector yields 0.784, with loss and capacity held fixed. The mechanism is the tie structure: with 93.0% of rows tied at the maximum, the argmax label is an arbitrary representative of a large tie class, and the projection onto one class discards the relative ordering the continuous target preserves. Two boundaries apply: the finding is judge-specific (the rejudge audit drops the tied rate to 20%; Appendix D), and recovering the prior band is not beating it: no tested reader uses the preserved information to exceed the prior (Section 4.3).

4.5 The opportunity set is prior-relative

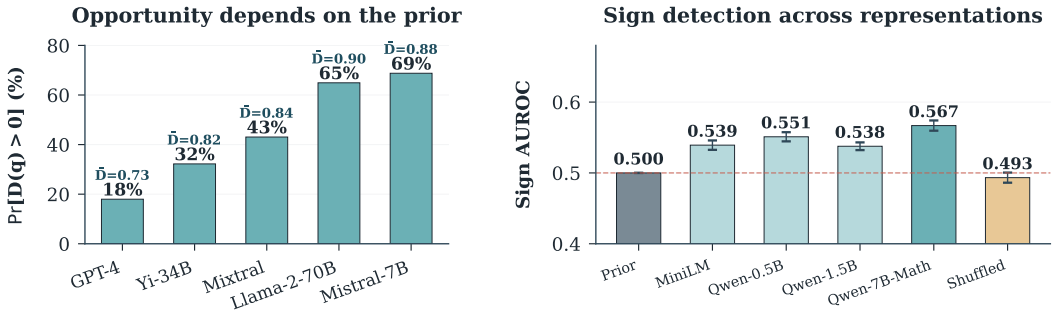


Figure 3: **Left:** the decision-bearing share under five priors on the fixed oracle; the opportunity set is a property of the prior. Bars show $\Pr[D(q) > 0]$; annotations give the mean gap on the $D > 0$ subset. **Right:** gap-detection AUROC (row-level ranking of $\mathbf{1}[D(q) > 0]$) across representations (5-fold CV, mean±std). Frozen LLM hidden states lift sign detection from 0.539 to 0.567; the shuffled control collapses to 0.493.

$D(q)$ is prior-relative, and its sparsity moves with the prior (Figure 3, left): as the prior weakens from GPT-4 to Mistral-7B, the decision-bearing share rises from 18.1% to 68.8% (Mixtral-8x7B 43.1%), and the conditional prize grows alongside, from 0.729 to 0.884. A routing gain demonstrated against a weak prior may be a correction of the prior rather than a discovery of query-specific signal; router comparisons should therefore always state the prior, the positive-gap rate, and the conditional prize; the framework makes these three numbers first-class outputs.

5 Candidate-Conditioned Signal: Stronger Predictability, Same Boundary

The query-only results leave a precise question: does information revealed when a candidate *processes* the query support better gap detection than the query alone? This section tests the richest

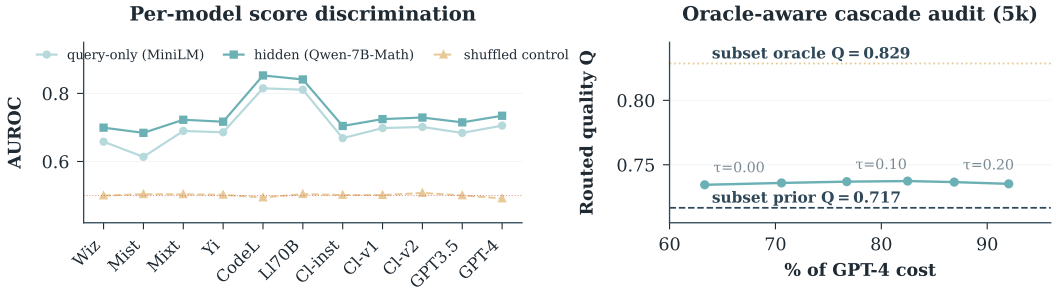


Figure 4: **Left:** score-discrimination AUROC (per query–model pair) under query-only (MiniLM) and LLM-hidden (Qwen-7B-Math) representations with the row-shuffled control (5-fold CV): the hidden representation improves all 11 candidates; shuffling destroys it. **Right:** oracle-aware cascade audit on the 5,000-query aligned subset vs. cost (threshold sweep, seed 7); an upper bound, not a deployable policy (Section 6).

query-side representations available, then the strictly stronger set $\mathcal{I}_{\text{cand}}$, using shuffle controls to attribute gains to alignment rather than capacity.

5.1 Representation quality governs sign detection

Replacing MiniLM embeddings with last-token hidden states of frozen Qwen2.5 encoders (0.5B, 1.5B, 7B-Math) changes what the *same linear reader* detects. Figure 3 (right) shows the gap-detection AUROC ladder across encoder scales, strongest for 7B-Math; the per-model ladder (Figure 4, left) shows score discrimination improving on all 11 candidates, with the largest lifts on the models hardest to predict from text. Two conclusions follow. The query-only bottleneck is not saturated at MiniLM. Yet the lift is bounded: the best representation reaches a gap-detection AUROC of 0.567 against an 18% base rate, and its routed policy stays inside the prior band ($Q = 0.744$, gain -0.037 ± 0.004; Appendix I). Stronger predictability did not becomeactionability: the reader still cannot identify *which* rows to switch.

5.2 Row-shuffle control: alignment, not capacity

An alternative explanation is that hidden states merely add dimensions. Shuffling the rows of the hidden-statematrix, which destroys the query–representation correspondence while preserving the marginal distribution, drops the control to gap-detection AUROC 0.493 and score-discrimination mean 0.501, chance level (Figure 4, left). The gain is therefore attributable to the *alignment* between each query and its representation, converting a correlation (bigger encoder, better AUROC) into a mechanism statement: the signal lives in the query-conditional representation, which is exactly the property candidate-conditioned routing must exploit.

5.3 Prefill probes: what the candidate's own states add

The set $\mathcal{I}_{\text{cand}}$ measures the strongest version of this alignment: for each candidate m, we extract the last-token *prefill* hiddenstate $h_m(q)$ (one forward pass, no generation) and fit a per-candidate probe predicting $s_m(q)$. On the 5,000-query aligned subset (three open candidates; Appendix F), probes reach routable-region routed quality of 0.678–0.826 across folds and candidates, against 0.24–0.30 for hard-label query-only readers on the same region; the lift concentrates on decision-bearing rows, consistent with theshuffle evidence: signal appears when the *pair* (query, candidate) is represented. Three boundaries apply. **(i) Scope:** prefill states exist only for open models, so $\mathcal{I}_{\text{cand}}$ routing is restricted to open pools. **(ii) Cost:** each candidate prefill costs a forward pass, so the only rational deployment is a two-stage design that consults $\mathcal{I}_{\text{cand}}$ selectively (Section 6). **(iii) Target-matching:** the probe comparison is not apples-to-apples with query-only readers (different target, subset, supervision), so we read the probes as an existence proof of interaction signal, not a head-to-head win. What they establish is that the information boundary is real and *located*: the missing signal for actionability is in the query–model interaction, measurable before generation.

What this does not establish. The probe results support three claims (interaction signal exists; it is stronger than any tested query-only representation; it is localized to decision-bearing rows) and explicitly do not support a fourth: that a deployable router can realize the corresponding gain. That gap is the subject of Section 6.

6 Diagnostic Headroom versus Selection-Valid Routing

The probe evidence motivates a natural architecture: a two-stage *decision-aware cascade* separating the cheap decision of *whether* to look from the expensive decision of *where* to route. Stage 1 (trigger) uses $\mathcal{I}_{\text{query}}$ to estimate whether the prior is beatable; Stage 2 (refinement) runs prefill probes on triggered queries and routes to the probe $\arg\max$ if it beats the trigger's prior prediction. We evaluate this design twice under two information protocols and show that the protocol, not the architecture, determines the answer.

6.1 Oracle-aware audit: an upper bound on trigger value

We first audit the cascade on the 5,000-query aligned subset under an oracle-aware diagnostic: the trigger signal is $\hat{D}_{\text{cheap}}(q) = \hat{s}_{m_0}(q) - s_{m_0}(q)$, using the prior's *observed score on the evaluated-query*, drawn from $\mathcal{I}_{\text{resp}}$ (illegal at deployment) and used here to measure how much value a trigger with perfect diagnostic information could realize. At $\tau = 0.10$ the audit reaches $Q = 0.7369$ at 77% of GPT-4 cost with a 25% trigger rate, against a subset prior of 0.7166 and oracle of 0.8286; the result is flat in $\tau \in [0.05, 0.30]$ and stable across five seeds (Figure 4, right; Appendix E). Read as a diagnostic, a trigger that could separate decision-bearing rows converts a meaningful fraction of the headroom at moderate cost; it says nothing about deployability. Read as a diagnostic: on this subset, a trigger that could separate decision-bearing rows would convert a meaningful fraction of the headroom at moderate cost. It says nothing about deployability.

Table 3: One cascade, two protocols: the oracle-aware audit and the leakage-free replay differ only in the trigger's test-time information (reference rows in Table 1).

Evaluation	Pool	Trigger inputs (test time)	Q	ΔQ	Trigger rate
Prior (reference)	full	—	0.7814	0	—
Oracle-aware audit	5k aligned	observed prior score	0.7369	+0.020	0.25
Leakage-free replay	full	query embedding only	0.7660	**−0.0154**	0.41

6.2 Leakage-free replay: the deployable answer

We then rerun the same pipeline under the selection-valid protocol: the trigger is a logistic reader on the query embedding trained against the training-fold label $\mathbf{1}[D(q) > 0]$, thresholds are selected by nested inner cross-validation, and the trigger's test-time inputs are the query embedding alone. On the full query pool, the replay reaches $Q = 0.7660$, *below* the 0.7814 prior, with gap-detection AUROC 0.6819 and a 40.8% trigger rate (Table 3). The mechanism is visible in the trigger's operating characteristic: 41% of rows are flagged, most of them ones where the prior was already optimal, and each such switch is a pure loss under Eq. equation 4. The base-rate hurdle is exactly quantifiable: with the empirical switch deltas of a random-candidate target ($+0.170$ per switch on decision-bearing rows, -0.356 on prior-optimal rows), a trigger must exceed *break-even precision* $0.356/(0.170 + 0.356) = 0.676$ to gain at all. The trigger's precision at useful recall (46–56%) is only 0.31–0.33, far below break-even, so its switches are net losses regardless of threshold; precision exceeds break-even only below 1% recall, where no headroom is recovered (Appendix F). The gap between 0.68 AUROC and the 0.676 precision requirement is the entire shortfall: ranking signal exists, but it is too weak to clear the bar at any useful operating point.

What would have to change for the trigger to recover the prior? Two levers are visible. The first is the trigger's precision at useful recall, which would need to clear break-even; at the observed base rate this requires a gap-detection AUROC far above anything we measure, even with candidate-conditioned 7B hidden states. The second is the base rate itself, which rises with a weaker prior: on Mistral-7B the decision-bearing share is 68.8% rather than 18.1%, so a trigger with even mediocre precision would beat the prior more often. Both are properties of the representation and the candidate

set, not of the trigger architecture; a routing evaluation that reports only a headline gain without the underlying headroom, base rate, and break-even leaves the failure mode invisible.

The contrast between the two rows is the paper's final empirical finding: *the same architecture is valuable under oracle-level diagnostics and harmful under deployment-legality*.

6.3 THE EVALUATION STANDARD

The results of Sections 4–6 define what a complete routing evaluation must report: **(1) headroom relative to a stated prior**, meaning Q_{prior}, Q_{oracle}, $\Pr[D(q) > 0]$, and the mean gap on decision-bearing rows; **(2) predictability** as R^2 or AUROC with the representation named; and **(3) selection-valid utility**: routed quality, switch rate, gain per switch, test-time inputs enumerated, oracle quantities excluded. A gain omitting (1) is uninterpretable, because a large number against a weak prior may only correct the prior; omitting (2) makes the claim unreproducible, because the same reader behaves differently under different representations; and omitting (3), unverifiable, while gains whose trigger inputs are not itemized may be oracle-aware without any sign of it in the headline. These are exactly the checks our own cascade fails under the naive protocol, which is why we report both rows. In practice the checklist costs one paragraph per experiment: the prior is a single line, the information set is a sentence naming the inputs, and the utility row is the same table the authors already produce—only with the oracle quantities moved to a separate, clearly labeled diagnostic section.

7 DISCUSSION, LIMITATIONS, AND CONCLUSION

What the results change. The decision gap turns the routing plateau into an auditable decomposition: the headroom is large in aggregate but lives on 18.1% of queries; query-only readers predict residual variation well but switch with near-zero conditional gain; hard labels lose the within-row structure that continuous targets preserve; candidate-conditioned representations expose the missing interaction signal; and the strictly legal cascade loses to the prior (Table 3). The diagnosis is a set of testable quantities, not a narrative; for *evaluation*, it implies three reporting requirements (prior, predictability, and selection-valid utility), and the cascade shows why the third is not optional: the replay looked like a -0.015 loss only after the protocol was enforced, never in the trigger's own metrics. For *the literature*, it explains why published routing gains often do not reproduce: headline numbers are reported against weak priors, with oracle-aware triggers, or under information sets the deployment cannot match, and the base-rate constraint is rarely stated. A reader who sees only the headline gain cannot tell whether it survives selection-validity; the gap makes the question testable on any score-matrix benchmark that releases its score matrix.

Limitations. Four boundaries define the scope of our claims. **(i) Benchmark:** conclusions concern the released RouterBench matrices, pool, judge, and cost accounting; tie-structure findings are judge-resolution-specific, and prior-relative statistics move with the pool. **(ii) Information sets:** candidate-conditioned evidence requires open models, three candidates, and a non-representative 5,000-query subset (Appendix F); probes are an information-boundary result, not a deployable method, and the alignment finding does not identify which deployable representation carries the same signal. **(iii) Prior family:** four alternative priors are ablated; ensembles, budget-conditioned objectives, and dynamic pools are not, though the prior-relativity result (Section 4.5) suggests the opportunity set changes with every portfolio change and should be re-measured per deployment rather than quoted from a benchmark. **(iv) Static scores:** offline judged scores, not live responses; deployment systems that query models directly see a different distribution of behavior.

Conclusion. The three quantities come apart: headroom of $+0.131$ on 18.1% of queries; readers flipping a third of decisions for near-zero gain; interaction signal only under candidate conditioning; and a cascade worth 0.737 under oracle-aware diagnostics but below the prior (0.766 vs. 0.781) under strict leakage-free evaluation. The standard is easy to state: report the headroom relative to a stated prior, the predictability, and the selection-valid utility—and let no router claim stand without all three.

Reproducibility statement

The main study uses the released RouterBench score matrix and MiniLM embeddings. Reader comparisons use shared stratified 5-fold cross-validation with seed 7. The 10-seed residual-gain table uses seeds 0–9. Per-candidate prefill states are extracted on a single 84 GB cloud GPU in fp16 or fp32. The score-matrix analysis and the leakage-free cascade replay run on CPU. The supplementary release provides a manifest, scripts, and unit tests; the release README documents the directory layout and reproduction commands.

Limitations

This study evaluates offline benchmark scores rather than live model responses. Results depend on RouterBench's model pool, judge granularity, and cost assumptions. Candidate-conditioned prefill states are available only for open models, and the strictly leakage-free cascade replay underperforms the prior; the headline cascade numbers are reported as oracle-aware analyses, not as a deployable end-to-end gain. We do not claim universal behavior across benchmarks or deployment APIs.

Ethics statement

The work uses public benchmark scores and public model checkpoints. It involves no human subjects, personal data, or sensitive attributes.

AI use statement

We used generative AI tools for polishing prose and providing code suggestions. We have not used generative AI tools for generating synthetic data, developing theoretical or conceptual frameworks, formulating mathematical claims, proving claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning or reformatting datasets, supporting data analysis, or interpreting results. We have reviewed all AI-assisted work: AI-suggested code was tested (65 unit tests) and the AI-suggested prose was manually edited for accuracy. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Acknowledgments

We thank the RouterBench team and the open-source community for releasing the benchmark and model libraries used in this study.

REFERENCES

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: how to use LLMs while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=cSimKw5p6R. Featured Certification.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs. *arXiv preprint arXiv:2410.10347*, 2025.

Yifan Dou, Shikan Lian, and Shibo Li. Conformal cascade: distribution-free accuracy guarantees for multi-tier LLM inference. *arXiv preprint arXiv:2607.25018*, 2026.

Tao Feng, Haozhen Zhang, Zijie Lei, Peixuan Han, and Jiaxuan You. GraphPlanner: Graph memory-augmented agentic routing for multi-agent LLMs. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=ZdGB7MNQDT.

Nigel Fernandez, Branislav Kveton, Ryan A. Rossi, Andrew S. Lan, and Zichao Wang. RADAR: Reasoning-ability and difficulty-aware routing for reasoning LLMs. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=CB6Ds5T4ae.

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. *arXiv preprint arXiv:2403.12031*, 2024.

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. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=ka82fvJ5f1.

Varun Kotte. UCCI: Calibrated uncertainty for cost-optimal LLM cascade routing. *arXiv preprint arXiv:2605.18796*, 2026.

Yang Li. Rethinking predictive modeling for LLM routing: when simple kNN beats complex learned routers. *arXiv preprint arXiv:2505.12601*, 2025.

Yifan Lu, Rixin Liu, Jiayi Yuan, Xingqi Cui, Shenrun Zhang, Hongyi Liu, and Jiarong Xing. RouterArena: An open platform for comprehensive comparison of LLM routers. In *International Conference on Learning Representations (ICLR)*, 2026a. URL https://openreview.net/forum?id=9HsaIi4ngF.

Yifan Lu, Qiyue Zhang, Shenrun Zhang, Zhibo Yu, Zhuang Wang, Hanjie Chen, and Jiarong Xing. The routing plateau: understanding and breaking the accuracy limits of LLM routers. *arXiv preprint arXiv:2606.07587*, 2026b.

Rafid Mahmood. Routing, cascades, and user choice for LLMs. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=VqAhhF6av8.

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. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Edoardo Pona, Milad Kazemi, Mehran Hosseini, Yali Du, David Watson, Osvaldo Simeone, and Nicola Paoletti. Calibrate-Then-Delegate: safety monitoring with risk and budget guarantees via model cascades. *arXiv preprint arXiv:2604.14251*, 2026.

Hang Zheng, Hongshen Xu, Yongkai Lin, Shuai Fan, Lu Chen, and Kai Yu. DiSRouter: Distributed self-routing for LLM selections. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=KDcwXKr0NU.

Wenhao Zheng, Yixiao Chen, Weitong Zhang, Souvik Kundu, Yun Li, Zhengzhong Liu, Eric P. Xing, Hongyi Wang, and Huaxiu Yao. CITER: collaborative inference for efficient large-language-model decoding with token-level routing. In *Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=nqX9UYW9Af.

Appendix

TABLE OF CONTENTS

A Data, Pool, and Judge

B Complete Reader Results

C Multi-Seed Analysis

D Judge-Resolution Audit

E Cascade Protocol Audit

F Candidate-Conditioned Probes

G Regional Decomposition

H Prior Ablation

I Representation and Capacity Controls

J Deployment Baselines and Cost Accounting

K External Pool (LLMRouterBench)

L Excluded Evidence and Registry Policy

A Data, Pool, and Judge

Score matrices. We use the released RouterBench 0-shot matrix (36,497 queries $\times$ 11 models) as the primary evaluation and the 5-shot matrix (36,511 $\times$ 11) as the replication split. The candidate pool, in released column order: WizardLM-13B-V1.2, Mistral-7B-Chat, Mixtral-8x7B-Chat, Yi-34B-Chat, CodeLlama-34B-Chat, Llama-2-70B-Chat, Claude-Instant-v1, Claude-v1, Claude-v2, GPT-3.5-Turbo-1106, GPT-4-1106-Preview. Scores are judged correctness in $[0, 1]$ on the released five-bin grid; missing entries are set to -1 and excluded from policy comparisons (a missing score is never counted as a routable alternative nor as a switch loss).

Prior definition. The global prior is $m_0 = \arg\max_m \mathbb{E}_{q \in \text{train}}[s_m(q)]$ computed on training folds only. On both splits this selects GPT-4-1106-Preview (0-shot full-pool prior quality 0.7814; 5-shot 0.8040). Selecting the prior on training folds avoids the leakage of a full-pool strongest-model prior; on this benchmark the selection is stable across folds, so the two definitions coincide numerically here.

Query features. MiniLM-L6 embeddings (384-d), released with the benchmark, are the default query representation throughout Section 4. Alternative representations (frozen Qwen2.5 encoders at 0.5B/1.5B/7B-Math scale) are used only in Section 5 and Appendix I, where they are explicitly named; no result mixes representations without labeling.

Judge granularity. The released judge is coarse: 93.0% of 0-shot rows are tied at the row maximum. Section D quantifies what a finer judge changes and why the tie-structure findings are judge-resolution-specific.

B Complete Reader Results

Table 4: Full reader grid on the 0-shot pool (routed quality; stratified 5-fold CV, seed 7). Continuous targets recover the prior band; hard labels collapse. "k-agg" is retrieval aggregation over training-fold neighbors.

Reader	Target	Q
KNN-20	hard argmax	0.519
LogReg-PCA64	hard argmax	0.510
MLP-CE (256–64)	hard argmax	0.522
MLP-BCE (256–64)	binarized correctness	0.523
MLP-MSE (256–64)	correctness vector	0.541
MLP-SOFT (256–64)	softmax correctness	0.784
Ridge (α=1)	correctness	0.782
k-agg@100	correctness (retrieval)	0.784
PairRank-MSE	pairwise ranking	0.780
IRT (2PL)	correctness	0.779
Global prior	—	0.781
Oracle	—	0.912

Training details. All readers are fit inside training folds. MLPs use one hidden architecture (256–64) across all targets so that target effects are not confounded with capacity; the PCA-64 projection for LogReg is fit on training folds only; $k \in \{1, 3, 5, 10, 20, 50, 100\}$ is tuned by inner 25% validation on each training fold; the IRT reader is a two-parameter logistic model fit by maximum likelihood. Random Forest and GBDT regressors were also run and land inside the Ridge band (± 0.002); they are omitted from tables for space. All readers share the same folds, so paired differences across readers are computed on identical row partitions.

Table 5: Target vs. loss isolation on the identical MLP backbone (256–64, 0-shot, seed-7 80/20). "CE" = cross-entropy on the tie-broken argmax label; "ML" = multilabel BCE on binarized correctness; "soft" = softmax-normalized correctness vector. "tie" credits any predicted model attaining the row maximum.

	CE (hard)		ML (multilabel)		Soft	
Region	exact	tie	exact	tie	exact	tie
Overall	0.509	0.528	0.523	0.542	**0.784**	**0.838**
Trivial	0.726	0.717	0.728	0.718	0.832	0.856
Routable	0.293	0.256	0.326	0.289	0.856	0.861
Strong-fail	0.060	0.602	0.060	0.600	0.028	0.028

The isolation is exact: same architecture, same folds, same capacity—only the target changes, and routed quality moves from 0.51 to 0.78. Two details sharpen the mechanism. First, the routable-region collapse under hard labels (0.29 vs. 0.86 soft) is where within-tie ordering matters most, consistent with the 93% tie rate. Second, the "tie" scoring column recovers only +0.02 for hard labels: the hard-label deficit is not a scoring artifact—it persists when any tied-correct prediction is credited. Source: `results/results_r52.json`.

C Multi-Seed Analysis

The ten-seed sweep uses independent 80/20 splits with seeds 0–9 (`numpy.default_rng`). Per-seed routed gains ($\times 10^{-3}$):

- **0-shot:** 3.56, 0.38, 2.47, 2.29, 0.03, 1.89, −2.88, 1.53, −0.16, 0.91; mean $+0.0010 \pm 0.0017$; flips $33.99\% \pm 0.66\%$; residual R^2 0.1337 ± 0.0047.

- **5-shot:** 1.08, 0.69, −1.31, 1.62, 1.00, −1.07, 0.08, 0.21, −0.81, −0.14; mean $+0.0001 \pm 0.0009$; flips $25.02\% \pm 0.93\%$; residual R^2 0.1385 ± 0.0047.

Eight of ten 0-shot seeds and six of ten 5-shot seeds are positive; the 95% confidence interval of the mean gain includes zero on both splits. The per-seed decision-event rate (share of rows where the reader's argmax differs from the prior and the row is decision-bearing) is $18.2\% \pm 0.3\%$ on 0-shot—the reader acts on decision-bearing rows at close to their base rate, confirming that the failure is discrimination quality, not avoidance of the decision-bearing subset. Source: `results/results_r59.json`.

D Judge-Resolution Audit

Two audits bound the tie-structure finding.

Rebinning (in-silico). Collapsing the released scores to coarser grids leaves the tied-best rate essentially unchanged: 93.0% (5-bin) → 93.9% (3-bin) → 94.5% (2-bin) on 0-shot, and 94.7% → 95.6% → 96.2% on 5-shot. The reason is that 91.4% of 0-shot rows already have binary $\{0, 1\}$ scores, and the remaining rows' coarse scores map to the same winning sets. The tie rate is therefore a property of judged binary correctness at the per-query level, not of the five-bin grid itself.

Continuous rejudging. Two hundred stratified queries were rejudged on a 0–100 scale (2,200 model judgments; judge correlation with released scores 0.715). The tied-best rate drops from 92% to 20%, and the mean tie size falls from 9.7 to 2.5 models.

Table 6: Continuous-rejudging audit by region (200 stratified queries). The routable region contains zero sampled rows, so no routable-region conclusion can be drawn from this audit.

Region	n	Orig. tied	Rescored tied	Δ
Trivial	185	0.962	0.195	−0.767
Routable	0	—	—	—
Strong-fail	15	0.400	0.267	−0.133
Overall	200	0.920	0.200	−0.720

Regional composition of the sample: 185 trivial, 15 strong-fail, 0 routable—so the audit supports the judge-resolution caveat only; it does not license region-level conclusions under a finer judge, and in particular the routable-region share under continuous judging remains unmeasured. The trivial-region drop (0.962 → 0.195) is the direct measurement behind the paper's judge-resolution caveat: most "tied" rows are ties of the judge, not ties of the models. Source: `path2_final_results.json`.

E CASCADE PROTOCOL AUDIT

Table 7: Oracle-aware cascade: complete seed × threshold sweep on the 5,000-query aligned subset. Q is flat in τ within ± 0.004 across all 30 cells; the trigger rate and cost move as expected.

Seed	Statistic	τ=0	0.05	0.10	0.15	0.20	0.30
7	Q	0.7345	0.7359	0.7369	0.7373	0.7366	0.7352
	trigger	0.40	0.32	0.25	0.19	0.14	0.09
	cost/GPT-4	0.63	0.71	0.77	0.82	0.87	0.92
0	Q	0.7318	0.7355	0.7378	0.7376	0.7372	0.7354
	trigger	0.41	0.32	0.25	0.18	0.14	0.08
	cost/GPT-4	0.63	0.70	0.77	0.83	0.87	0.92
1	Q	0.7307	0.7342	0.7356	0.7348	0.7351	0.7339
	trigger	0.40	0.33	0.25	0.18	0.14	0.09
	cost/GPT-4	0.64	0.70	0.77	0.83	0.87	0.92
2	Q	0.7329	0.7352	0.7365	0.7373	0.7368	0.7349
	trigger	0.39	0.32	0.25	0.18	0.14	0.09
	cost/GPT-4	0.64	0.70	0.77	0.83	0.87	0.92
3	Q	0.7319	0.7354	0.7365	0.7363	0.7368	0.7352
	trigger	0.40	0.32	0.25	0.19	0.14	0.09

Oracle-aware audit (5k aligned), full sweep and seeds. The audit uses trigger signal $\hat{D}_{\text{cheap}}(q) = \hat{s}_{m_0}(q) - s_{m_0}(q)$, where s_{m_0} is the prior's *observed* score on the evaluated query ($\mathcal{I}_{\text{resp}}$ input). It measures attainable headroom under perfect diagnostic information, not deployable performance. Source: `results/cascade_robustness/tau_robustness.json`.

Leakage-free replay (full pool). Trigger: logistic regression on MiniLM embeddings, label $\mathbf{1}[D(q) > 0]$, class-balanced, C=1; outer stratified 5-fold CV (seed 7); per-outer-fold threshold selected by inner 5-fold CV maximizing balanced accuracy on the outer-training rows; refinement uses prefill probes of the three open candidates on triggered rows only, routing to the probe argmax when its prediction exceeds the query-only prior prediction.

Table 8: Leakage-free replay, full pool (36,497 queries).

Quantity	Value
Routed Q	0.7660
Prior Q	0.7814
ΔQ	-0.0154
Trigger AUROC	0.6819
Trigger rate	0.408
Per-fold thresholds	0.492, 0.516, 0.543, 0.492, 0.509 (median 0.516)

Why the replay loses. The trigger's sign detection is genuinely informative (AUROC 0.68), but the positive-gap base rate is 18.1%: at the inner-CV-selected threshold, 40.8% of rows are flagged, so a large majority of flagged rows are prior-optimal. Each harmful switch is a pure loss under Eq. equation 4, and the per-switch conditional gain cannot absorb the false-trigger mass. Raising the threshold trades recall for precision but cannot win: precision at 80% recall is only 0.24 (Appendix F), far below what break-even requires at this base rate. The replay's test-time inputs are the query embedding alone; observed scores, oracle gaps, and judge outcomes are forbidden inputs, enforced in code and asserted by unit tests (`test_r103_leakage_free.py`).

Reproducibility note. A superseded manifest recorded $Q = 0.7201$ for this replay; the executable implementation reproducibly yields 0.7660 (re-verified during the rewrite), the release manifest was corrected, and the registry (`paper/evidence_registry.json`) freezes 0.7660 as canonical. Source: `code/r103_leakage_free_cascade.py`; result `results/r103_leakage_free_cascade.json`.

Table 9: Canonical candidate prefill arrays: shapes and SHA-256 prefixes (full hashes in the release manifest).

Array	Shape	SHA-256 (first 12)
mistral	36497x1x4096	da931a63a6d4
mixtral	36497x1x4096	7de6491e9a65
yi	36497x1x7168	066422a2cb7e
qwenmath	36497x1x3584	143bfe799d07

F CANDIDATE-CONDITIONED PROBES

Aligned subset and its representativeness. The 5,000-row aligned subset (first 5,000 0-shot rows with all three open candidates scored) is *not* pool-representative:

Table 10: Aligned-subset vs. full-pool composition. KS tests reject equality of the $D(q)$, prior-score, and oracle-score distributions ($p < 10^{-8}$).

Quantity	5k aligned	Full pool
$\Pr[D(q) > 0]$	0.221	0.181
Mean D on $D > 0$	0.506	0.721
Prior Q	0.717	0.781
Oracle Q	0.829	0.912
Knowledge share	0.294	0.735
Math share	0.363	0.205
Coding share	0.000	0.012

Table 11: Trigger precision–recall operating points for detecting $\mathbf{1}[D(q) > 0]$ (query-only features). At matched recall, pool precision is far below subset precision—the quantitative reason the full-pool replay loses to the prior.

	5k aligned		Full pool	
Target recall	achieved	precision	achieved	precision
0.5	0.47	0.44	0.50	0.32
0.6	0.57	0.41	0.60	0.29
0.7	0.67	0.39	0.70	0.26
0.8	0.80	0.37	0.80	0.24
0.9	0.90	0.34	0.90	0.22

Trigger calibration and the break-even precision. The class-balanced trigger used in the replay is trained against the 18.1% positive base rate, so its raw scores are systematically inflated; out-of-fold Brier is 0.224 with expected calibration error 0.274. Post-hoc isotonic calibration within each outer training fold reduces Brier to 0.139 and ECE to 0.005 (Figure 5, middle), i.e., the trigger becomes a genuinely calibrated probability of "prior is beatable." Calibration alone does not change the operating trade-off: the break-even precision for a switch policy whose target is a random other candidate is 0.676—computed from the empirical expected switch deltas ($+0.170$ on decision-bearing rows, -0.356 on prior-optimal rows)—while the trigger's precision at useful recall (46–56%) is 0.31–0.33. Precision only exceeds break-even at recall below 1%, where it recovers no headroom. This is the quantitative statement of the base-rate hurdle: the detector must either move far up the precision axis or the switch target must be far better than random, and the candidate-conditioned probes of Section 5 are the measured route to the latter.

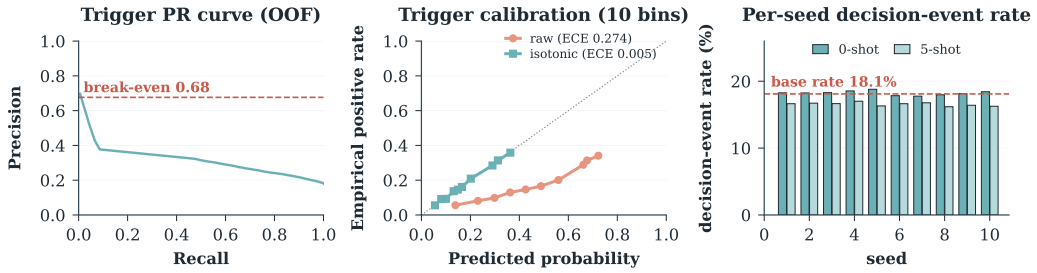


Figure 5: Leakage-free trigger diagnostics (out-of-fold, full pool). **Left:** precision–recall curve for detecting $\mathbf{1}[D(q) > 0]$; the dashed line is the break-even precision for a random-candidate switch policy (0.676). **Middle:** reliability curves before and after within-fold isotonic calibration. **Right:** per-seed decision-event rate of the residual reader against the decision-bearing base rate; the reader acts on decision-bearing rows at close to their natural rate, confirming that the failure is discrimination, not avoidance.

(lower prior, lower conditional prize, no coding), all subset numbers are labeled subset-only in the main text and the full-pool replay of Appendix E is the deployment-relevant measurement. Source: `results/r90_5k_representativeness.json`.

Trigger detection quality on subset vs. pool. Binary detection of $\mathbf{1}[D(q) > 0]$ with query-only features: AUROC 0.782 on the aligned subset vs. 0.683 on the full pool; AUPRC 0.498 vs. 0.314. The precision–recall sweep shows the operating tension: at 80% recall the trigger's precision is 0.375 (subset) vs. 0.241 (pool)—most flagged rows are prior-optimal, which is exactly the failure mode the full-pool replay realizes. Source: `results/r90_5k_representativeness.json`.

Probe recipe. For each candidate, last-token prefill state from the final transformer layer (Mistral-7B bf16, Mixtral-8x7B int8, Yi-34B bf16; single 84 GB GPU, batch sizes 2–16 by model size). Ridge (α=100) on standardized states within stratified 5-fold CV; Mixtral uses a two-layer MLP head. Per-fold routable-region routed quality:

Table 12: Per-candidate prefill probes: overall and routable-region routed quality by fold (aligned subset, 5-fold CV). The fold-wise global prior is shown for reference.

Candidate	Region	f1	f2	f3	f4	f5	Mean
Mistral-7B (linear)	overall	0.821	0.834	0.828	0.828	0.808	0.824
	routable	0.826	0.752	0.678	0.722	0.716	0.739
Mixtral-8x7B (MLP)	overall	0.733	0.730	0.778	0.696	0.714	0.730
	routable	0.807	0.777	0.739	0.700	0.788	0.762
Yi-34B (linear)	overall	0.689	0.606	0.693	0.628	0.642	0.651
	routable	0.783	0.588	0.845	0.761	0.732	0.742
Global prior	overall	0.823	0.832	0.828	0.828	0.828	0.828

Overall vs. routable. The overall routed quality of the probes (0.70–0.83 across folds) sits at or above the training-fold prior on the same subset, while the routable-region quality spans a wide band. The probes are therefore strongest precisely where model identity matters; their overall advantage is bounded by the trivial mass, exactly as the decision-gap view predicts.

Fold-to-fold variance is large (e.g., Yi 0.588–0.845), consistent with the small routable mass per fold; we therefore report the range in the main text and the per-fold values here. For reference, the training-fold global prior on the same region is ≈ 0.83. Source: `probe_results_{mistral,mixtral,yi}.json`.

Cost sensitivity of the audit. The audit's Q is invariant to the assumed prefill/decode cost ratio: sweeping $r \in \{0.2, 0.3, 0.5, 0.7, 1.0, 1.3\}$ leaves $Q = 0.7278$ fixed while total cost moves only from 64.4% to 66.1% of GPT-4 cost, because the trigger rate—not the per-candidate cost—dominates the cost structure at these ratios. Source: `results/r90_5k_representativeness.json` (part D).

G Regional Decomposition

Table 13: Decision-gap anatomy by descriptive region (0-shot, seed-7 80/20 test rows; $D(q)$ relative to the GPT-4 prior). Regions are computed from observed outcomes and are diagnostic only.

Region	n	Prior Q	Oracle Q	$\Pr[D > 0 \mid \text{region}]$
Trivial	3,926	0.824	0.936	0.176
Routable-hard	2,425	0.891	1.000	0.110
Routable-mid	427	0.665	0.746	0.290
Strong-fail	250	0.040	0.917	0.924
Chaos (pool-unanswered)	272	0.000	0.000	0.000

The strong-fail region concentrates decision opportunity ($\Pr[D > 0] = 0.924$) but is small (3.4% of rows) and mostly unanswered by the whole pool (oracle 0.917 reflects the one model that occasionally succeeds). The routable-mid region has the highest per-region positive rate among solvable regions (0.290) at small mass (4.3%). Trivial rows dominate the pool (53.8% of test rows) and contribute 52.7% of all decision-bearing rows under the GPT-4 prior—a consequence of the coarse judge, under which many "trivial" rows still admit a strictly better candidate. Source: `results/r80_prior_ablation.json`.

H Prior Ablation

Full-pool statistics under five priors (0-shot; oracle fixed):

Table 14: Prior ablation. The positive-gap rate and the conditional prize grow together as the prior weakens; both are first-class outputs of the framework (Section 6.3).

Prior	$\Pr[D(q) > 0]$	Mean D on $D > 0$	Mean D (all rows)
GPT-4	0.180	0.729	0.131
Yi-34B	0.322	0.816	0.263
Mixtral-8x7B	0.430	0.845	0.364
Llama-2-70B	0.649	0.901	0.585
Mistral-7B	0.688	0.884	0.608

Table 15: Prior ablation by region (0-shot, seed-7 test rows): per-region prior quality, shared oracle quality, and the within-region positive-gap rate. Region definitions are oracle-based, so the oracle row is constant across priors.

Prior	Statistic	trivial	rout.-hard	rout.-mid	strong-fail	chaos
GPT-4	prior Q	0.824	0.891	0.665	0.040	0.000
Yi-34B	prior Q	0.715	0.703	0.489	0.041	0.000
Mixtral-8x7B	prior Q	0.704	0.406	0.424	0.256	0.000
Llama-2-70B	prior Q	0.372	0.300	0.425	0.040	0.000
Mistral-7B	prior Q	0.533	0.000	0.246	0.043	0.000
Oracle (shared)	oracle Q	0.936	1.000	0.746	0.917	0.000
GPT-4	$\Pr[D{>}0]$	0.18	0.11	0.29	0.92	0.00
Yi-34B	$\Pr[D{>}0]$	0.29	0.30	0.60	0.92	0.00
Mixtral-8x7B	$\Pr[D{>}0]$	0.31	0.60	0.70	0.70	0.00
Llama-2-70B	$\Pr[D{>}0]$	0.64	0.70	0.69	0.93	0.00
Mistral-7B	$\Pr[D{>}0]$	0.49	1.00	1.00	0.92	0.00

The per-region view adds the mechanism behind the aggregate: under the GPT-4 prior, decision opportunity concentrates in strong-fail ($\Pr[D{>}0] = 0.92$) and routable-mid (0.29), while trivial and routable-hard rows are nearly prior-optimal. Under weak priors the positive rate saturates everywhere (Mistral prior: routable-hard 1.00, trivial 0.49), showing that a weak prior converts even easy rows into routing opportunities—another reason prior-relative reporting is mandatory. Figure 6 renders the full matrix.

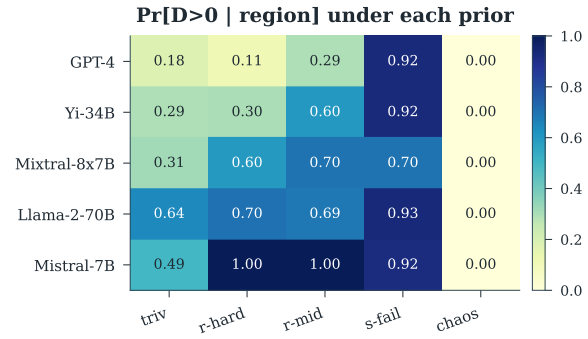


Figure 6: Within-region positive-gap rate $\Pr[D(q) > 0]$ for each prior–region pair (0-shot). Opportunity moves from the strong-fail corner (GPT-4 prior) toward saturation across all regions as the prior weakens.

Two observations. First, the decision-bearing share spans $18\% \rightarrow 69\%$ across plausible deployment priors, so any routing claim that omits the prior is uninterpretable. Second, the conditional prize saturates near 0.88–0.90 for weak priors (bounded by the score scale), while for the strongest prior the conditional prize (0.729) is still large—the plateau is therefore not a small-prizes problem under any tested prior. Source: `results/r80_prior_ablation.json`.

I Representation and Capacity Controls

Encoder ladder (query-side, routed quality). Frozen Qwen2.5 encoders as query representations, same Ridge reader and folds:

Table 16: Query-side representation ladder (0-shot, 5-fold CV). Predictive quality and routed utility move in different directions across encoders.

Representation	Raw R^2	Routed Q	Gain vs. prior	Flip rate
MiniLM-L6 (384-d)	0.129	0.7814	$+0.0001$	—
Qwen2.5-0.5B (896-d)	0.139	0.7725	-0.0089 ± 0.0017	0.353
Qwen2.5-1.5B (1536-d)	0.113	0.7616	-0.0198 ± 0.0022	0.422
Qwen2.5-Math-7B (3584-d)	0.001	0.7442	-0.0372 ± 0.0040	0.542

Scaling the tested frozen encoders does not recover actionability; the larger encoders switch *more* and lose more. The 7B-Math representation is simultaneously the best sign detector (AUROC 0.567) and a below-prior router—a clean demonstration that predictability and actionability are different quantities. Source: `results/8b_reader/results.json`, `results/encoder_ablation_partial.json`.

Gap-detection ladder (sign AUROC for $\mathbf{1}[D(q) > 0]$). Prior 0.500; MiniLM 0.539 ± 0.007; Qwen-0.5B 0.551 ± 0.006; Qwen-1.5B 0.538 ± 0.005; Qwen-7B-Math 0.567 ± 0.007; shuffled 0.5B 0.493 ± 0.007. R^2 on $D(q)$ follows the same ordering with small magnitudes (0.029 MiniLM, 0.051 Qwen-7B-Math). Source: `results/decision_gap/decision_gap_results.json`.

Per-model AUROC ladder. Mean per-model score AUROC: query-only MiniLM 0.703; Qwen-0.5B hidden 0.728; Qwen-1.5B hidden 0.724; Qwen-7B-Math hidden 0.739; row-shuffled 0.501. The full 11-model breakdown is plotted in Figure 4 (left). The largest per-model lifts appear on Claude-v1/v2 and GPT-4 (+0.03–0.05), the candidates whose correctness is hardest to predict from text features. Source: `results/decision_gap/per_model_auroc_ladder.json`.

J Deployment Baselines and Cost Accounting

GP-Gate (query-only escalation gate). A mid-tier-default cascade (Mixtral default; GPT-4 escalation) gated by a calibrated query-only confidence reader reaches $Q = 0.691$ at \$0.00091/query on the full pool; a FrugalGPT-style k-NN gate reaches $Q = 0.621$ at \$0.00059/query. Both sit below the GPT-4 prior ($Q = 0.7814$ at \$0.00327/query) in quality at much lower cost. Consistent with the framework: these gates exploit the prior's average dominance rather than identifying decision-bearing rows, so their gains are cost-side, not decision-side—they improve the cost–quality Pareto position without closing any of the decision gap.

Table 17: Canonical per-query cost accounting (released dollar costs; prefill = prefix cost where applicable). GPT-4 vs. cheapest probe candidate: $35\times$ dollars, $6.7\times$ FLOPs, $\approx 4\times$ latency. The superseded $200\times$ claim is excluded.

Item	$/query	Share of GPT-4
GPT-4-1106 (prior, decode)	$0.00329	100.0%
GPT-4-1106 (prefill only)	$0.00165	50.0%
Mixtral-8x7B (decode)	$0.00014	4.1%
Mixtral-8x7B (prefill)	$0.00007	2.0%
Yi-34B (decode)	$0.00019	5.6%
Yi-34B (prefill)	$0.00009	2.8%
Mistral-7B (decode)	$0.00005	1.4%
Mistral-7B (prefill)	$0.00002	0.7%
Always GPT-4 policy	$0.00327	99.3%
GP-Gate policy	$0.00091	27.6%
FrugalGPT-style gate	$0.00059	17.9%
Static Ridge reader	$0.00020	6.1%
Audit cascade (τ=0.10, 5k)	$0.00251	76.2%

Table 18: Master comparison on the full 0-shot pool: every deployment-relevant policy on one surface. Query-only readers cluster at the prior; gates trade quality for cost; the oracle is the bound.

Policy	Q	$/query
Always GPT-4 (prior, no routing)	0.7814	$0.00327
Ridge correctness (continuous)	0.7819	$0.00020
K-agg@100 (retrieval, Hu 2024)	0.7840	$0.00020
PairRank-MSE (ranking, Hu 2024)	0.7800	$0.00020
KNN-20 argmax (Hu 2024)	0.5190	$0.00010
LogReg-PCA64 argmax (Hu 2024)	0.5100	$0.00010
MLP-CE argmax (Hu 2024)	0.5220	$0.00010
MLP-MSE correctness (Hu 2024)	0.5410	$0.00020
GP-Gate tau=0.00 (ours)	0.6910	$0.00091
FrugalGPT k=20 tau=0.40	0.6210	$0.00059
Oracle (per-query best)	0.9121	$0.00028

Cost ratios. The defensible GPT-4-vs-probe-candidate ratios are $35.4\times$ in dollars, $6.7\times$ in FLOPs, and $\approx 4\times$ in latency; an earlier "$200\times$" claim is excluded from the evidence registry as an accounting error. Prefill cost assumptions affect only the audit's cost axis (Appendix F); replay and reader results are cost-free measurements of quality. The master table makes the deployment trade-off explicit: query-only readers cluster at the prior's quality at 6–16% of its cost, escalation gates trade quality for cost without closing the decision gap, and no tested policy joins high quality with low cost—the empty upper-left region is the plateau in cost–quality coordinates. Source: `results/r102_master_table.json`, `results/cost_analysis_honest.json`.

K External Pool (LLMRouterBench)

A regenerated external replication on LLMRouterBench aggregates (19 tasks with $\geq$12 aligned models, mean 22.8 models/task): mean prior $Q = 0.500$, mean oracle $Q = 0.747$, mean gap 0.246. The aggregate headroom is larger than RouterBench's (0.246 vs. 0.131), consistent with weaker average models in the pool. The per-task table shows the same structural pattern as RouterBench: hard tasks (AIME $0.46 \to 0.97$, GPQA $0.74 \to 1.00$, HLE $0.34 \to 0.55$) carry large priors-to-oracle gaps, while saturated tasks (BBH $0.83 \to 0.99$, HumanEval $0.88 \to 0.99$) compress headroom—the external analogue of the prior-relativity finding: opportunity lives where the pool's mean is far from its ceiling.

Table 19: External pool (LLMRouterBench aggregates): per-task model count, mean-model performance (prior proxy), and best single model (oracle proxy). Aggregate: mean prior 0.500, mean oracle 0.747, mean gap 0.246 over 19 tasks. Per-query decision gaps are not computable on this dataset (no per-query scores), so this table is directional evidence only.

Task	$\|M\|$	Prior	Oracle	Task	$\|M\|$	Prior	Oracle
gpqa	38	0.74	1.00	mbpp	20	0.73	0.94
winogrande	20	0.77	1.00	livecodebench	38	0.78	0.92
kandk	20	0.79	1.00	simpleqa	14	0.62	0.90
bbh	20	0.83	0.99	finqa	20	0.71	0.88
humaneval	22	0.88	0.99	meld	20	0.61	0.88
math500	20	0.92	0.99	livemathbench	35	0.78	0.85
medqa	20	0.79	0.98	korbench	20	0.62	0.78
arcc	20	0.85	0.97	emorynlp	20	0.46	0.76
mmlupro	36	0.83	0.97	hle	15	0.34	0.55
aime	38	0.46	0.97				

We report this as directional consistency only: the dataset lacks per-query scores, so per-query decision gaps, tie structure, and reader replication cannot be computed, and earlier headline claims from this source (task counts and prior definitions varying across artifacts) are excluded from the evidence registry. Source: `results/lrb_replication_full.json`.

L Excluded Evidence and Registry Policy

The evidence registry (`paper/evidence_registry.json`) records what this paper does and does not cite. Excluded artifacts: the oracle-aware *full-pool* cascade ($Q \approx 0.828$; trigger uses the observed test-query prior score); the conflicting manifest replay value (0.7201; provenance unrecoverable, superseded by the reproducible 0.7660); the LLMRouterBench headline replication (conflicting task counts and prior definitions; Appendix K replaces it); stacked region shares from an overlapping-count JSON; a bootstrap CI whose stored mean lies outside its own interval; and figures with hard-coded values superseded by the data-driven regeneration scripts shipped in the release. Every number in the main text traces to a named JSON in the registry, and the release unit tests assert the core quantities.