CoRoute: Learning Routing Frontiers as Ordered Policy Hulls

Anonymous authors
Paper under double-blind review

Abstract

A routing service exposes several operating points at once, and callers expect them to agree: a query accepted at a lower tier should stay accepted when coverage or compute allowances increase. Independent per-allowance optimization, however, offers no such guarantee: a more generous tier can withdraw a query that a stricter tier accepted. We treat the routing frontier as the object to learn, rather than one operating point, and require the learned family to preserve acceptance across tiers. For a shared score and a finite library of threshold-action policies, CoRoute learns tier policies within an ordered policy hull, using first-order stochastic dominance to couple their threshold distributions. One shared draw nests realized acceptance for every input while leaving model selection flexible. The order condition is linear, enabling globally optimal joint fitting within the library. Its fitted-risk cost is zero exactly when independently optimal policies admit a coherent selection. As a reusable policy layer, CoRoute keeps router predictors and LLMs fixed and adds only tens of microseconds once scores are available. Independent service data can additionally certify coverage and bounded resource budgets. Experiments on published LLM cascades and multiple routing architectures establish withdrawal and evaluate the component. In an independent external evaluation, CoRoute reduces the mean fraction of requests losing service on at least one upgrade from about 8% to zero at a mean accuracy cost of 0.04 percentage points, while reducing the largest tier losses relative to a strong single-cut alternative.

1 Introduction

LLM routers select models under inference budgets (Hu et al., 2024; Dekoninck et al., 2025). Multi-tier services must also decide what an upgrade preserves. A higher tier can answer more requests overall yet decline a request served at a lower tier. We call this failure *service withdrawal*.

Independently optimized tiers may serve different request sets while meeting their own quality, coverage and budget objectives. Consider the held-out SimpleQA query, "Who won Season 2 Snatch Game on RPDR?" The lower tier accepts it and selects DeepSeek-R1. With a larger budget and the same coverage goal, the independently fitted higher tier declines this same request before invoking an LLM (Figure 1). After adding CoRoute, the request remains accepted at the higher tier, which selects Gemini 2.5 Pro (Figure 2).

We learn a *routing frontier*, a family of concurrent service-tier policies with a fixed scoring interface. It provides *persistent acceptance*: accepted requests remain accepted at higher comparable tiers, while the router may select a different LLM.

Withdrawal can be substantial. In the adapted-router study, more than a quarter of test requests lose service on some upgrade in the most affected split. This is an individual-request failure that aggregate coverage can hide. Whenever acceptance probability decreases, some withdrawal is unavoidable unless the policies change. We also observe it in a published abstaining cascade using its original policies (Section 4.2).

CoRoute reuses a router's score and model-selection rules under coverage, budget and acceptance-order constraints. Predictors and LLMs stay fixed; deployment adds hashing and sampling from small policy tables.

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.

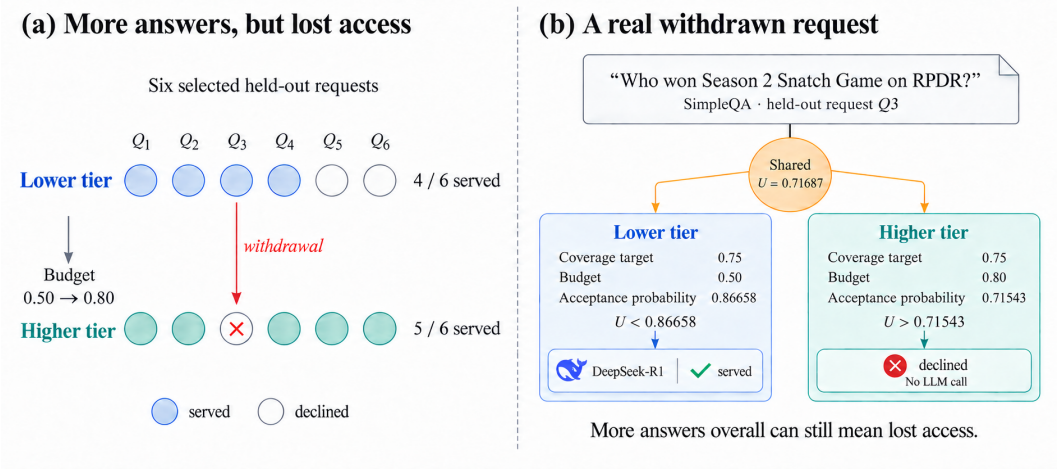

Figure 1: A larger budget can withdraw a previously served request. (a) Filled circles mark service for six selected held-out requests. (b) The same SimpleQA request uses one shared draw $U \in [0, 1)$ at both tiers. A tier serves it when U is below its acceptance probability. Here that probability falls at the higher tier, causing withdrawal. Coverage targets concern the overall workload; budgets are normalized. These selected examples illustrate the failure, not its frequency.

We derive exact acceptance-order constraints and minimize the largest fitted accuracy cost of persistence within a finite routing library. The theory identifies when persistence is free and when joint fitting strictly improves on fitting tiers in sequence.

Experiments test reuse across four routing architectures, withdrawal and tier protection on six external LLM tasks, and service certification on a separate five-task, 20-model workload. Acceptance-preserving controls assess the policy layer; a tierwise-fitting ablation isolates joint optimization.

2 Service withdrawal across LLM tiers

A selective LLM service either declines a request x or chooses a model to answer it. Tier $k \in \{1, \dots, K\}$ promises a minimum answered fraction $\gamma_k > 0$ and sets a mean inference budget β_k. Service requirements specify either individual tasks or a fixed workload mixture. The quality objective can protect each task in a set $\mathcal{E}$; a pooled workload or task-specific cascade uses one risk group. The router does not observe task identity at deployment. Let $I_k(x)$ indicate whether the request is accepted.

A tier ℓ is higher than k, written $k \preceq \ell$, when $\gamma_k \le \gamma_\ell$ and $\beta_k \le \beta_\ell$. Both coordinates may increase, or just one. Tiers with opposing changes are incomparable. Let E contain the adjacent ordered pairs; enforcing order on these pairs suffices by transitivity. **Persistent acceptance** requires a joint realization of the policies such that $I_k(x) \le I_\ell(x)$ for every request and every $(k, \ell) \in E$, almost surely over their shared randomness.

Why independent fitting cannot ensure the contract. Write $p_k(x)$ for tier k's acceptance probability. Any coupling with these marginals obeys

$$\Pr(I_k = 1, I_\ell = 0 \mid x) \ge [p_k(x) - p_\ell(x)]_+. \tag{1}$$

Joint acceptance has probability at most $\min(p_k, p_\ell)$, which gives the bound. A common uniform draw, accepting when $U < p_k(x)$, attains it for every pair simultaneously. Hence a shared random draw alone cannot fix a probability reversal. The tier policies must be fitted to respect acceptance order. This lower bound holds for any router; the shared-score construction below makes zero withdrawal efficient to enforce.

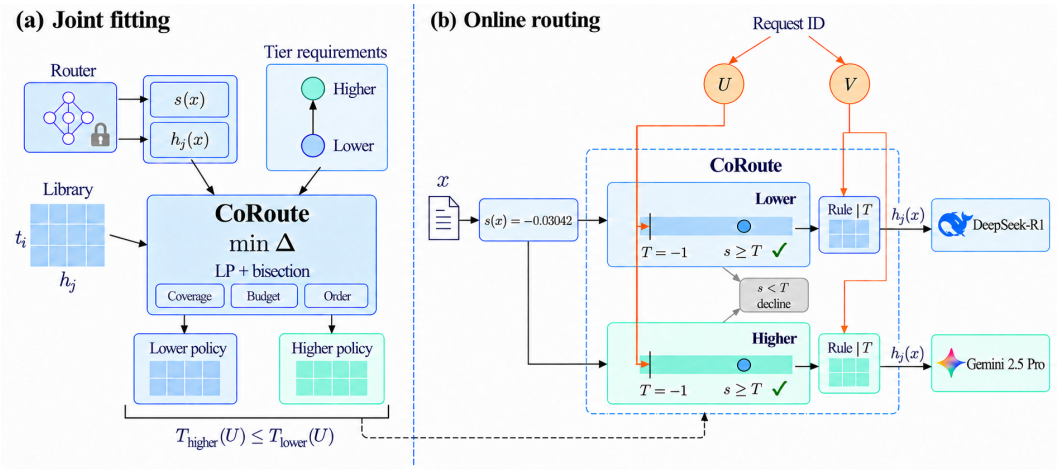

Figure 2: (a) The fixed router supplies acceptance score $s(x)$ and model-selection rules h_j. Library cells index threshold–rule pairs; policy grids contain their weights w_{kij}. Joint fitting minimizes Δ, the largest fitted-risk increase over independent tier optima, under coverage, budget and acceptance order. The tier arrow raises coverage or budget requirements. (b) Both tiers reuse request-ID draws U, V: U selects threshold $T = T_k(U)$; if $s(x) \ge T$, V selects a rule conditional on T, and $h_j(x)$ returns the model ID. Gray branches decline. One invocation uses one tier; colors and grids are schematic. Steps and example: Algorithm 1 and Section J.1.

3 CoRoute: A Policy Layer for Existing Routers

CoRoute takes a shared acceptance score $s(x)$, deterministic routing rules $h_j(x) \in \{1, \dots, A\}$ over A LLMs, and service tiers. The score ranks requests for acceptance; each rule selects an LLM. A fixed router supplies quality–cost rules, with constant-model rules also available (Section B.1). CoRoute changes their mixture weights, leaving predictors fixed.

The layer separates *whether to answer* from *which model answers*. Mixing thresholds spreads acceptance across score ranges; conditional routing allocates models within them. Only acceptance is ordered across tiers. A single-cut control tests the value of this flexibility.

3.1 Mixing Routing Rules Without Withdrawing Accepted Requests

Choose ordered thresholds $t_1 < \cdots < t_I$, including always-accept and always-decline options. Each candidate policy (i, j) accepts if $s(x) \ge t_i$ and, when accepted, invokes $h_j(x)$. Tier k assigns nonnegative weights w_{kij} that sum to one over this finite set, which we call the routing library. These mixtures form the *policy hull*. Constraining tier-specific mixtures to preserve acceptance across tiers defines an *ordered policy hull*. Tier k's threshold marginal, the total probability of selecting threshold i, is $u_{ki} = \sum_j w_{kij}$, and its acceptance probability at score s is $A_k(s) = \sum_{i:t_i \le s} u_{ki}$.

Theorem 3.1 (Ordered thresholds give persistent acceptance)**.** *For an adjacent pair $(k, \ell) \in E$, $A_k(s) \le A_\ell(s)$ at every score if and only if*

$$\sum_{i \le r} u_{ki} \le \sum_{i \le r} u_{\ell i} \quad \textit{for every threshold prefix } r. \tag{2}$$

Under these constraints, one uniform U shared across tiers samples thresholds $T_k(U)$ with $T_\ell(U) \le T_k(U)$. Accepting when $s(x) \ge T_k(U)$ therefore preserves acceptance for every request. Conditional routing rules may differ.

The higher tier assigns at least as much probability to every prefix of easier thresholds: finite-support first-order stochastic dominance (Strassen, 1965). The guarantee includes unseen requests and is independent of model count and routing-rule complexity (proof and the role of score support in Section A).

3.2 FITTING ACCURACY AND SERVICE REQUIREMENTS TOGETHER

We fit the tiers to make persistence as inexpensive as possible. Let $z_a(x) \in [0,1]$ be model a's error, $b(x) \geq 0$ the cost of computing the shared score, and $c_a(x) \geq 0$ the additional answering cost, with finite means. Three quantities describe each policy on task e: coverage C, mean accepted error per arriving request N, and mean cost per request G:

$$C_{ke} = \mathbb{E}_e[I_k], \qquad N_{ke} = \mathbb{E}_e[I_k z_{a_k}], \qquad G_{ke} = \mathbb{E}_e[b + I_k c_{a_k}]. \tag{3}$$

The error among accepted requests is $R_{ke} = N_{ke}/C_{ke}$; service requires $C_{ke} \geq \gamma_k$ and $G_{ke} \leq \beta_k$. We protect the worst task so that easy tasks cannot hide poor answers on a harder one (Sagawa et al., 2020). Direct-routing experiments use $b = 0$ and normalized model costs, with router overhead measured separately. For a cascade, b includes the first model call even if the request is declined.

For each candidate policy, we estimate coverage and accepted-error statistics $(C^{\mathrm{fit}}_{em}, N^{\mathrm{fit}}_{em})$ on labeled fitting data. Separate calibration data supply coverage and cost statistics $(C^{\mathrm{cal}}_{em}, G^{\mathrm{cal}}_{em})$ without outcome labels. Here $m = (i,j)$ indexes a threshold–rule pair. These statistics, also called moments, are linear in the weights. Calibration rows g may specify individual tasks or a fixed mixture. The service-feasible set is

$$\mathcal{P}_k = \{w \geq 0 : \ \mathbf{1}^\top w = 1, \quad C^{\mathrm{cal}}_g w \geq \gamma_k + \epsilon_C, \quad G^{\mathrm{cal}}_g w \leq \beta_k - \epsilon_G \quad \forall g\}. \tag{4}$$

Affine rows can also impose exact calibration coverage and positive fitting coverage. The margins allow for sampling variation; an independent service sample determines whether a fitted family can be certified.

First compute each tier's independent optimum, $r^\star_k = \min_{w \in \mathcal{P}_k} \max_e N^{\mathrm{fit}}_e w / C^{\mathrm{fit}}_e w$. This is the lowest fitted risk with the same library, statistics and service constraints, but without requiring persistence. Thus $1 - r^\star_k$ is an upper bound on fitted worst-task accuracy in this fixed configuration. Adding persistence can match this bound or incur a cost. CoRoute minimizes that cost:

$$\Delta^\star = \min_{\substack{w_k \in \mathcal{P}_k \ (k=1,\dots,K) \\ \text{subject to } equation\ 2}} \max_k \left[\max_e \frac{N^{\mathrm{fit}}_e w_k}{C^{\mathrm{fit}}_e w_k} - r^\star_k \right]. \tag{5}$$

Thus $\Delta^\star$ is the smallest fitted-risk increase at the most affected tier, not a test-accuracy bound. Simultaneous fitting retains the risk cap, calibration requirements and acceptance order in the deployed weights.

Theorem 3.2 (Exact fitting within the routing library)**.** *Suppose the joint feasible set is nonempty and fitting coverage is strictly positive throughout each $\mathcal{P}_k$. At a fixed cap Δ, the risk constraints are the linear rows*

$$[N^{\mathrm{fit}}_e - (r^\star_k + \Delta) C^{\mathrm{fit}}_e] w_k \leq 0. \tag{6}$$

Together with Eqs. (2) *and* (4)*, these form a linear-program (LP) feasibility problem, monotone in Δ. Bisection returns the global optimum of Eq.* (5) *up to numerical tolerance. Every comparator using the same library, moments, service constraints and prefix-order constraints has a fitted maximum risk increase at least as large as $\Delta^\star$.*

All fits check coverage positivity and use the protocol's fixed tie-break (Sections A and B).

What joint fitting coordinates.

Proposition 3.3 (Acceptance at equal coverage)**.** *For two ordered tiers on the same workload, let $A_k(x) \leq A_\ell(x)$ be their acceptance probabilities. Then*

$$\mathbb{E}|A_\ell - A_k| = \mathbb{E}A_\ell - \mathbb{E}A_k.$$

Equal coverage therefore forces $A_k = A_\ell$ almost surely. Fitting the lower tier first commits both tiers to that acceptance function on the workload.

Changing models cannot undo this commitment. Joint fitting instead chooses acceptance for the whole family. This distinction can matter strictly: Section A.7 constructs two tiers with identical coverage and routing libraries where tierwise fitting incurs maximum excess risk 0.20, while the joint optimum is 0.12. The construction and proof are in Section A.7; transferring fitted gains to test quality also depends on risk-estimation accuracy (Section A.6).

Algorithm 1 CoRoute: offline fitting and online routing

Input: fixed score s, rules $\{h_j\}$, thresholds $t_1 < \cdots < t_I$; fitting/calibration statistics, tier requirements, order E; tolerances $\varepsilon_{\text{opt}} > 0$, $\varepsilon_{\text{tie}} \geq 0$.

Offline: fit one policy family

1. Check each $\mathcal{P}_k$ is feasible with positive fitting coverage; compute $r_k^\star$.
2. Set $L = 0$, $H = 1$. Return infeasible if the LP at H is infeasible.
3. **While** $H - L > \varepsilon_{\text{opt}}$:
   $d \leftarrow (L + H)/2$; solve the LP with Eqs. (2), (4) and (6) at cap d.
   If feasible, $H \leftarrow d$; if infeasible, $L \leftarrow d$.
4. At cap $H + \varepsilon_{\text{tie}}$, apply the fixed tie-break under all LP constraints.
5. Store $u_{ki} = \sum_j w_{kij}$ and $\pi_{kij} = w_{kij}/u_{ki}$ when $u_{ki} > 0$.

Online: request x at tier k

6. Derive $U, V \in [0, 1)$ from the request ID and fixed key using separate hash streams; reuse both draws across tiers.
7. $i \leftarrow \min\{r : \sum_{a \leq r} u_{ka} > U\}$.
8. **If** $s(x) < t_i$, **return** decline.
9. $j \leftarrow \min\{r : \sum_{b \leq r} \pi_{kib} > V\}$.
10. **Return** model ID $h_j(x)$ for caller dispatch.

LP solver errors abort fitting. Freeze the selected configuration before independent certification; if requested, deploy only a family passing Proposition A.3.

The performance cost of persistence. Persistence is free exactly when independent optimal policies can be chosen compatibly. Otherwise, searching near-optimal policies finds the smallest fitted-risk allowance that removes all conflicts (Propositions A.1 and A.2). This distinguishes poor selection among equally good policies from an unavoidable performance cost.

Certifying service requirements. Fix the candidate families before drawing an independent i.i.d. service sample, and assume per-request cost lies in $[0, B]$ for a known bound B. Simultaneous coverage lower bounds and cost upper bounds account for every candidate, tier and prespecified sample size. With probability at least $1-\delta$, any family passing all these bounds meets every population coverage and expected-cost requirement (Proposition A.3). No answer labels are needed; the guarantee concerns the specified deployment workload.

Deployment interface. Algorithm 1 gives the offline fit and online adapter. Compared tiers share the score, policy version, request identifier and randomization key. Order holds for each shared draw; fitted mixture quality and fixed-hash deployment quality are evaluated separately (Section J).

4 Experiments

4.1 Evaluation setup

We use 10,539 aligned records from four tasks in the official LLMRouterBench response-cost release (Li et al., 2026a): HLE, LiveCodeBench, MMLU-Pro and SimpleQA (Phan et al., 2025; Jain et al., 2025; Wang et al., 2024; Wei et al., 2024). Training costs select five LLMs spanning the cost range in each split. All methods share these models and normalized archived benchmark costs.

We fix BGE query embeddings (Xiao et al., 2024) and use four predictors: a small MLP, RouteLLM-style matrix factorization, EmbedLLM-style TextMF and RouterDC-style utilities (Ong et al., 2025; Zhuang et al., 2025; Chen et al., 2024b). These adaptations share a five-model interface and remain fixed across policy methods. Three MLP splits and six per adapted architecture give 21 cases, each with nine coverage–budget tiers. Validation selects each method's rule library and risk estimator. Candidate libraries use constant-model rules alone or add query-dependent quality–cost rules; acceptance always uses $s(x)$. Section B gives model pools, adaptations, data roles and selection grids.

Comparators and ablation. *Independent* fits tiers separately under their service constraints. Its fitted optimum bounds what is possible in the same library without persistence; its test accuracy is

an empirical reference, not an upper bound. *Single-cut mixtures* use one ordered acceptance boundary per tier and mix model-selection rules. *Single model + abstention* chooses one model per tier. Boundary randomization allows both controls to match calibration coverage exactly. Implementation details are in Section B.

The *tierwise-fitting ablation* replaces only CoRoute's joint solve, keeping its library, constraints and shared deployment. It freezes each fitted tier before fitting successors. Validation chooses between two legal orders.

Evaluation. Our main accuracy metric is $100(1-\max_e R_{ke})$, averaged over the nine tiers. We call it *worst-task accuracy*; HLE is the worst task throughout. Macro accuracy averages both tasks and tiers. To measure tier protection, *max. tier loss* is the largest test accuracy loss against Independent within a case; we report its mean across cases. Actual cost and coverage are reported alongside accuracy because common targets need not yield identical operating points.

We separate two withdrawal measures. *Probability drop* is $100 \max_{(k,\ell)\in E} \sup_s [A_k(s) - A_\ell(s)]_+$, in percentage points. *Withdrawal rate* is the percentage of test requests losing service on at least one adjacent upgrade, using a shared draw per normalized prompt. Accuracy, cost and coverage are computed exactly from mixture weights and cached model outcomes. Throughout, "points" means percentage points (pp).

Paired 95% intervals use 2,000 bootstrap resamples of prompt groups within each split. Splits overlap and are not independent replications. This corpus was used in development; the external confirmation below uses separate reserved prompts (Sections B and F).

4.2 Withdrawal in an existing abstaining LLM cascade

We replay the original Early Abstention policies (Zellinger et al., 2025) on six tasks and 16 model pairs, without a CoRoute fit. An upgrade increases the abstention penalty or relaxes the cost penalty. Training data select adjacent pairs with nondecreasing coverage and cost, and 50–95% coverage at both ends. Withdrawal occurs in 79 of 92 eligible cases, affecting up to 11.1% of requests on one upgrade: a later model can decline a previously answered query. Section E gives complete counts and eligibility rules.

4.3 Persistent service across router architectures

Table 1: Four architectures on 10,539 records and four tasks: 21 five-model configurations, nine tiers each. Values average tiers and splits. Accuracy, coverage and withdrawal are percentages; costs are normalized. Withdrawal counts requests losing service on any adjacent upgrade. Pass counts tiers meeting every task's test coverage and budget requirements. Bold marks zero withdrawal. Splits share requests.

	Accuracy (%)					
Policy	**Macro**	**Worst task**	**Cost**	**Coverage**	**Withdrawal**	**Pass**
MLP (3 splits; 27 tier–split points)						
Independent	56.16	17.64	0.433	90.65	13.07	27/27
+ CoRoute	56.13	17.56	0.434	90.52	**0.00**	27/27
RouteLLM-style (6 splits; 54 tier–split points)						
Independent	56.36	18.27	0.447	81.30	6.35	54/54
+ CoRoute	56.37	18.26	0.447	81.00	**0.00**	54/54
EmbedLLM-style (6 splits; 54 tier–split points)						
Independent	56.37	18.46	0.441	88.25	15.83	54/54
+ CoRoute	56.57	18.24	0.437	89.01	**0.00**	54/54
RouterDC-style (6 splits; 54 tier–split points)						
Independent	56.30	18.33	0.443	81.59	9.85	51/54
+ CoRoute	56.26	18.21	0.440	81.87	**0.00**	51/54

Table 1 reports the complete before–after comparison for each architecture. CoRoute eliminates withdrawal while keeping macro and worst-task accuracy close to Independent; actual cost and coverage show the operating points. Per-task accuracy is in Table 16.

All budgets pass; three coverage misses occur on one RouterDC task–split. Tierwise fitting passes every point. All cases are included. Empirical service checks are separate from the structural acceptance guarantee.

Cost of adding the component. With router scores available, median policy-selection latency is 30.9–32.7 microseconds per tier across six profiled external cascades. Fitting is offline; the component retrains neither the router predictor nor the LLMs. This timing excludes score computation and LLM calls (Sections F and J).

4.4 Routing quality under comparable service conditions

Table 2: Matched calibration coverage on all 21 router–split cases, using the same full library and nominal budgets. Accuracy is mean worst-task accuracy (%); max. loss is relative to zero-margin Independent (pp). Cost and coverage (fraction) are test means. Pass counts jointly satisfied service points out of 189. Bold marks the best displayed accuracy or loss among persistent methods.

Method	Accuracy	Max. loss	Cost	Coverage	Pass
Independent (zero margin)	19.99	0.00	0.488	0.802	69/189
Independent	19.58	0.85	0.464	0.802	183/189
Single-cut mixtures	19.22	2.10	0.462	0.802	180/189
Single model + abstention	15.70	8.06	0.250	0.802	187/189
CoRoute (tierwise fit)	19.51	1.18	0.464	0.802	187/189
CoRoute	**19.52**	**1.17**	0.464	0.801	179/189

Table 2 matches calibration coverage exactly under common budgets, fixed predictors and the full library. Service constraints apply to the task-balanced workload; quality still protects the worst task. Actual test cost and coverage remain visible.

CoRoute improves worst-task accuracy and reduces maximum tier loss relative to single-cut mixtures; both paired intervals exclude zero, at almost identical test coverage and similar actual resource use. Independent with a validation-selected budget margin isolates the small additional mean accuracy cost of imposing persistence. The zero-margin row shows why good fitted accuracy alone does not establish reliable service.

Coordination and the cost of preserving access. With population risks known, all 24 multi-task configurations benefit from joint fitting: mean maximum tier loss falls from 1.19 to 0.44 points. This controlled finite-population diagnostic establishes a coordination benefit, not a held-out improvement. With estimated risks, matched test intervals against tierwise fitting include zero; Section D isolates the role of sample size.

Across all 21 original fits, even the best choice among independent optima leaves acceptance conflicts. Yet every conflict disappears with less than 0.50 points of fitted-risk allowance (Figure 4 and Section C). Nearby policies can differ sharply in who receives service while having similar quality. CoRoute uses this freedom to preserve access at small cost.

4.5 Independent confirmation on external LLM cascades

We freeze policies and primary comparisons before opening reserved Early Abstention data from six tasks and 16 model pairs. Using the first model's confidence, CoRoute declines the request, returns its answer or calls the second model. Every first call is charged, including declines. The native cascade retains later-model abstention.

The *Native cascade* uses thresholds fitted to our service targets. *Union repair* preserves any answer available from its lower tiers.

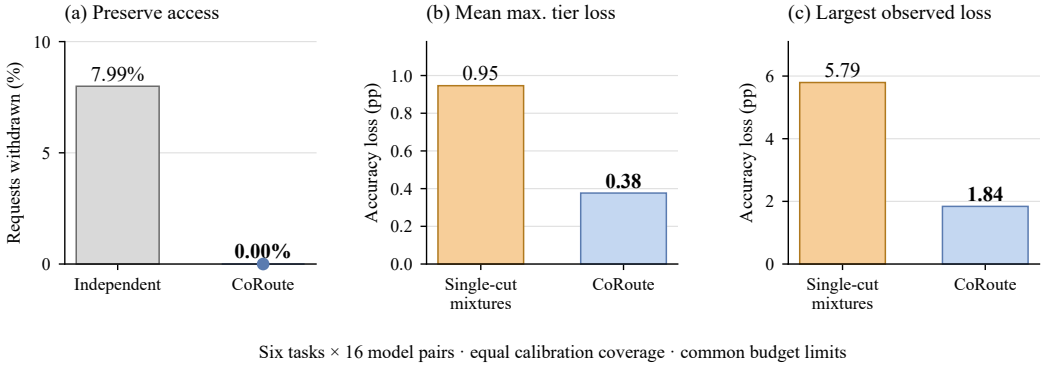

Figure 3: Independent confirmation on 96 external cases. (a) Case-average withdrawal on any adjacent upgrade. (b) Mean maximum tier loss against Independent. (c) Largest observed tier loss, not a confidence bound. All axes start at zero. Panels (b,c) compare single-cut mixtures with CoRoute; Table 3 includes accuracy, cost, native controls and the tierwise-fitting ablation.

Table 3: External confirmation at equal calibration coverage and common budgets: 96 task–cascade cases, nine tiers each. Accuracy (%), cost and coverage (fraction) are means. Max. tier loss averages each case's largest loss against Independent (pp). Withdrawal counts requests losing service on any adjacent upgrade (case-average %). Pass counts joint test coverage–budget passes out of 864. Bold marks the best mean among persistent methods, not statistical superiority.

Method	Accuracy	Max. tier loss	Cost	Coverage	Withdrawal	Pass
CoRoute	76.30	**0.38**	0.451	0.701	0.00	778
Independent (reference)	76.34	0.00	0.452	0.701	7.99	762
Single-cut mixtures	**76.33**	0.95	0.445	0.702	0.00	748
Native cascade	75.74	3.14	0.433	0.700	20.47	796
Union repair	75.00	3.75	0.527	0.742	0.00	417
Optimization ablation						
CoRoute (tierwise fit)	76.32	**0.38**	0.449	0.702	0.00	760

The nine tiers cross three coverage targets and three budgets. We test minimum and *equal calibration coverage*; randomized single-cut boundaries permit exact matching. Paired intervals resample prompts within tasks, jointly across model pairs. Cases share prompts (protocol: Section F).

CoRoute removes Independent's withdrawals at a mean accuracy cost of 0.039 points (95% interval $[0.018, 0.062]$). Compared with single-cut mixtures, mean maximum tier loss falls by 0.57 points ($[0.54, 0.90]$). A secondary task-level breakdown shows the same direction on all six tasks, so the reduction is not driven by one task (Section F). Tierwise fitting remains competitive; CoRoute meets both service constraints at 778 of 864 test points (Table 3).

Quality of the preserved service. In a secondary analysis of fixed policies, 70.8% of higher-tier answers retained by CoRoute but withdrawn by Independent meet the archived task's success criterion. CoRoute preserves its own accepted set; refitting need not retain every Independent answer. Outcome transitions and uncertainty are in Section F and Table 11.

4.6 Persistent access with certified resource budgets

We test a separate workload of five tasks and 20 models from the EmbedLLM release (Zhuang et al., 2025; Wikit, 2026). Predictor training, policy fitting, calibration, service certification and final testing use disjoint prompt groups. All methods share the same fixed embedding-based router,

coverage targets and resource limits. Released resource weights define a bounded cost proxy; quality uses the released fractional benchmark scores. This workload tests certification and model-pool reuse; Independent withdrawal is rare (0.084%).

Table 4: Independent confirmation on five new tasks with 20 models and nine simultaneously certified tiers. Score is expected selective benchmark score (%); max. loss is relative to certified Independent (pp). Cost and coverage evaluate fixed-hash deployment; coverage is a fraction. Cost is a bounded resource proxy, not an API price. Bold marks the best displayed score or loss among persistent methods.

Method	Score	Max. loss	Cost	Coverage	Certified
Independent (reference)	60.04	0.000	0.151	0.727	9/9
Single-cut mixtures	59.85	0.257	0.154	0.739	9/9
Single model + abstention	50.48	10.311	0.072	0.739	9/9
CoRoute (tierwise fit)	**60.04**	0.017	0.151	0.727	9/9
CoRoute	**60.04**	**0.006**	0.151	0.727	9/9

The selected CoRoute family passes simultaneous coverage and resource certification at every tier and meets those requirements on the final test sample. All accepted mass uses query-dependent routing rules, with 16 of the 20 models each receiving more than 1% of that mass; mean score stays close to Independent. Relative to single-cut mixtures, its reduction in maximum tier loss has a paired 95% interval of $[0.10, 0.92]$ points. Tierwise fitting remains close. Risk-estimation checks, deployed scores and sample-size sensitivity are in Section H.

5 Related work

LLM routing. FrugalML and FrugalGPT allocate model calls under inference budgets (Chen et al., 2020; 2024a). RouteLLM, EmbedLLM, RouterDC and cost-aware contrastive routing learn model-selection rules (Ong et al., 2025; Zhuang et al., 2025; Chen et al., 2024b; Shirkavand et al., 2025). OmniRouter, WISERouter and PROTEUS address workload and serving constraints (Mei et al., 2025; Li et al., 2026b; Bhatti et al., 2026). Abstaining cascades trade off error, cost and abstention (Zellinger et al., 2025). CoRoute preserves acceptance across service tiers.

Selective and ordered prediction. Selective prediction decides when to answer (Chow, 1957; El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017); deferral chooses an expert (Madras et al., 2018; Mozannar & Sontag, 2020; Verma et al., 2023). Nested decision regions and conformal sets impose ordering across levels (Lee & Scott, 2010; Gupta et al., 2022; Ochoa Rivera & Tewari, 2026). CoRoute orders acceptance under coverage and budget requirements while preserving model choice. Constrained randomized prediction supplies mixture-fitting tools (Narasimhan, 2018; Agarwal et al., 2018); we use them to quantify the quality cost of preserving access.

Model-update compatibility. Aggregate gains can hide errors on previously correct examples (Bansal et al., 2019; Srivastava et al., 2020). Positive-congruent training and MUSCLE reduce such regressions during model updates (Yan et al., 2021; Echterhoff et al., 2024). CoRoute keeps models fixed and preserves acceptance across service levels. Table 11 measures the separate question of answer-quality regressions.

6 Discussion and conclusion

CoRoute adds persistent acceptance to existing LLM routers without retraining. Across router architectures and external cascades, it preserves access at small mean accuracy cost and reduces tier losses relative to single-cut mixtures.

Joint fitting solves the fitted problem optimally; its test benefit depends on policy conflicts and risk estimates. Our ablations do not establish a stable advantage over tierwise fitting. Stronger conflicts, better risk estimates and guarantees beyond a fixed score or for answer quality remain open directions.

Ethics statement. We use public benchmarks without recruiting participants. Persistent acceptance guarantees neither answer safety nor correctness; constraints across known tasks do not establish fairness for unobserved groups.

AI USE STATEMENT

The authors developed the ideas, designed experiments and wrote the initial manuscript. AI assisted literature searches, polishing, review of experiments and result interpretations, figure refinement, and implementation under author-defined protocols. Authors reviewed this assistance and remain responsible. Quantitative figures use scripts and recorded results.

REPRODUCIBILITY STATEMENT

Proofs, protocols and deployment details appear in Sections A, B, G, H and J. The supplement supplies code, frozen policies, results, figure scripts, randomization settings and data preparation instructions.

REFERENCES

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, 2018. URL https://proceedings.mlr.press/v80/agarwal18a.html.

Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S. Weld, Walter S. Lasecki, and Eric Horvitz. Updates in human-AI teams: Understanding and addressing the performance/compatibility tradeoff. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):2429–2437, 2019. doi: 10.1609/aaai.v33i01.33012429. URL https://ojs.aaai.org/index.php/AAAI/article/view/4087.

Amit Singh Bhatti, Vishal Vaddina, and Dagnachew Birru. PROTEUS: SLA-aware routing via Lagrangian RL for multi-LLM serving systems. *arXiv preprint arXiv:2601.19402*, 2026. URL https://arxiv.org/abs/2601.19402.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalML: How to use ML prediction APIs more accurately and cheaply. In *Advances in Neural Information Processing Systems*, volume 33, pp. 10685–10696, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/789ba2ae4d335e8a2ad283a3f7effced-Abstract.html.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024a. URL https://openreview.net/forum?id=cSimKw5p6R.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *Advances in Neural Information Processing Systems*, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7a641b8ec86162fc875fb9f6456a542f-Abstract-Conference.html.

C. K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, 1957. doi: 10.1109/TEC.1957.5222035.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs. In *International Conference on Machine Learning*, pp. 12987–13010, 2025. URL https://proceedings.mlr.press/v267/dekoninck25a.html.

Jessica Maria Echterhoff, Fartash Faghri, Raviteja Vemulapalli, Ting-Yao Hu, Chun-Liang Li, Oncel Tuzel, and Hadi Pouransari. MUSCLE: A model update strategy for compatible LLM evolution. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7320–7332, 2024. doi: 10.18653/v1/2024.findings-emnlp.430. URL https://aclanthology.org/2024.findings-emnlp.430/.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010. URL https://jmlr.org/papers/v11/el-yaniv10a.html.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, 2017. URL https://papers.neurips.cc/paper_files/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Chirag Gupta, Arun K. Kuchibhotla, and Aaditya Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *Pattern Recognition*, 127:108496, 2022. URL https://doi.org/10.1016/j.patcog.2021.108496.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

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

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/94074dd5a072d28ff75a76dabed43767-Paper-Conference.pdf.

Gyemin Lee and Clayton Scott. Nested support vector machines. *IEEE Transactions on Signal Processing*, 58(3):1648–1660, 2010. doi: 10.1109/TSP.2009.2036071.

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. LLMRouterBench: A massive benchmark and unified framework for LLM routing. *arXiv preprint arXiv:2601.07206*, 2026a. URL https://arxiv.org/abs/2601.07206.

Yifei Li, Zihui Gao, and Laks V. S. Lakshmanan. WISERouter: LLM routing with workload budget constraint. *arXiv preprint arXiv:2607.23765*, 2026b. URL https://arxiv.org/abs/2607.23765.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/09d37c08f7b129e96277388757530c72-Abstract.html.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. In *Conference on Learning Theory*, 2009. URL https://arxiv.org/abs/0907.3740.

Kai Mei, Wujiang Xu, Minghao Guo, Shuhang Lin, and Yongfeng Zhang. OmniRouter: Budget and performance controllable multi-LLM routing. *arXiv preprint arXiv:2502.20576*, 2025. URL https://arxiv.org/abs/2502.20576.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html.

Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *AISTATS*, 2018. URL https://proceedings.mlr.press/v84/narasimhan18a.html.

Eduardo Ochoa Rivera and Ambuj Tewari. Online conformal prediction: Enforcing monotonicity via online optimization. *arXiv preprint arXiv:2605.12668*, 2026. URL https://arxiv.org/abs/2605.12668.

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 from preference data. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025. URL https://arxiv.org/abs/2501.14249.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1911.08731.

Reza Shirkavand, Shangqian Gao, Peiran Yu, and Heng Huang. Cost-aware contrastive routing for LLMs. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e46eb6403af68506331f941282d838aa-Abstract-Conference.html.

Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. An empirical analysis of backward compatibility in machine learning systems. *arXiv preprint arXiv:2008.04572*, 2020. URL https://arxiv.org/abs/2008.04572.

Volker Strassen. The existence of probability measures with given marginals. *The Annals of Mathematical Statistics*, 36(2):423–439, 1965. doi: 10.1214/aoms/1177700153.

Rajeev Verma, Daniel Barrejon, and Eric Nalisnick. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In *International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 11415–11434, 2023. URL https://proceedings.mlr.press/v206/verma23a.html.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://doi.org/10.52202/079017-3018.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024. URL https://arxiv.org/abs/2411.04368.

Wikit. RoutingCompendium: performance and cost datasets, 2026. URL https://huggingface.co/datasets/Wikit/RoutingCompendium-perf. Versioned EmbedLLM split; revisions and checksums in supplementary material.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024. URL https://arxiv.org/abs/2309.07597.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14294–14303, 2021. doi: 10.1109/CVPR46437.2021.01407.

Michael J. Zellinger, Rex Liu, and Matt Thomson. Cost-saving LLM cascades with early abstention. *arXiv preprint arXiv:2502.09054*, 2025. URL https://arxiv.org/abs/2502.09054.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/bf5e4b85d203481d6e37bd32d9600162-Abstract-Conference.html.

A PROOFS AND FITTING DETAILS

A.1 ACCEPTANCE ORDER AND THE DEPLOYMENT COUPLING

For threshold marginal $u_{ki} = \sum_j w_{kij}$, write $F_k(t_r) = \sum_{i \le r} u_{ki}$, the cumulative distribution function (CDF) of its threshold. A library policy accepts exactly when its threshold does not exceed the score, independently of its routing rule. Thus $A_k(s) = F_k(s)$. The prefix inequalities in Eq. (2) imply $A_k(s) \le A_\ell(s)$ by choosing the largest threshold below s. Conversely, evaluating acceptance order on each threshold interval recovers every nontrivial prefix inequality. The total-mass prefix is equality. Thresholds at $-\infty$ and $+\infty$ encode always-accept and always-decline for finite scores.

For $U \sim \mathrm{Unif}(0,1)$, define the generalized inverse $T_k(U) = \inf\{t : F_k(t) \ge U\}$. Since $F_k \le F_\ell$, their inverses obey $T_\ell(U) \le T_k(U)$. Consequently $\mathbf{1}\{s(x) \ge T_k(U)\} \le \mathbf{1}\{s(x) \ge T_\ell(U)\}$ for every x. This proves Theorem 3.1. The same draw works on the entire tier order, including incomparable branches joined by multiple edges. Once threshold i is sampled, drawing rule j with probability w_{kij}/u_{ki} preserves the original policy weights. Zero-mass thresholds are never sampled; the conditional law there is immaterial. A second uniform V, independent of U, implements these conditional draws and may itself be shared across tiers.

The equivalence is between prefix constraints and order at every score. Input-level persistent acceptance implies all prefix constraints only when the score map reaches every relevant threshold interval. Our program enforces the stronger universal score condition and needs no reachability assumption. Because the proof marginalizes routing rules, increasing their number or internal complexity does not change the order block. Arbitrary bounded per-model costs and errors still have linear mixture moments.

Withdrawal severity and affected traffic. For an adjacent upgrade, the maximum acceptance-probability drop is

$$W_{k\ell} = \sup_s [A_k(s) - A_\ell(s)]_+ = \max_r \left[\sum_{i \le r} (u_{ki} - u_{\ell i}) \right]_+ . \tag{7}$$

The reported maximum is over adjacent edges. A maximum over all comparable pairs can be larger for a family that withdraws service; both vanish under persistent acceptance. For a fixed input, Eq. (1) is attained by the shared-uniform coupling, whose withdrawal interval is $[p_\ell(x), p_k(x))$ when nonempty. This minimizes every pairwise withdrawal simultaneously. It does not assert optimality of the union of all edge events over every possible coupling. Our traffic statistic evaluates that union under the specified shared coupling on attained test inputs, whereas $W_{k\ell}$ ranges over the score domain.

A.2 EXACTNESS OF JOINT SERVICE-CONSTRAINED FITTING

Each $\mathcal{P}_k$ is a closed subset of a finite-dimensional probability simplex and is therefore compact. If $C_e^{\mathrm{fit}} w > 0$ throughout it, the risk ratios and their finite maximum are continuous. Every independent minimum is attained. The joint set is also compact because its additional constraints are linear; nonemptiness ensures attainment of Eq. (5).

For positive denominators, $N_e^{\mathrm{fit}} w_k / C_e^{\mathrm{fit}} w_k \le r_k^\star + \Delta$ is equivalent to Eq. (6). The simplex, calibration service and prefix-order rows are linear as well. Increasing Δ weakens every risk row, so feasible caps form a closed upper interval starting at $\Delta^\star$. Independent optimal fitted risks can first be obtained by the same bisection without order constraints. Bisection on the joint cap then converges to the global finite-library optimum. Any comparator with persistent acceptance in this same set is feasible at its own maximum excess and hence upper-bounds $\Delta^\star$. This proves Theorem 3.2.

The positivity assumption can be checked from the two moment arrays before fitting. Define $d_e = \max_m (C_{em}^{\mathrm{cal}} - C_{em}^{\mathrm{fit}})$. A convex combination of these differences is at most d_e, so

$$C_e^{\mathrm{fit}} w = C_e^{\mathrm{cal}} w - (C_e^{\mathrm{cal}} - C_e^{\mathrm{fit}}) w \ge \gamma_k + \epsilon_C - d_e > 0. \tag{8}$$

This bound applies when calibration constrains the same task. For aggregate service constraints, the matched and model-pool studies explicitly impose $C_e^{\mathrm{fit}} w \ge 10^{-6}$ for every risk group. These affine rows ensure positive denominators throughout the feasible sets for all compared methods.

Implementation. In the router-reuse study, with K tiers, I thresholds and J rules, the program has KIJ mixture weights. The order block has $|E|(I-1)$ nontrivial rows. Our $K = 9$, $I = 23$ configuration has 1,035 weights for the constant-rule library or 3,105 for the full library. We use 28 bisection steps and LP primal/dual tolerances 10^{-9}. After the minimax fit, an L1 tie-break toward the independent family is restricted to a risk cap increased by $\varepsilon_{\text{tie}} = 10^{-5}$. This is a tie-break within the service-constrained problem, not a later service repair. A nonoptimal solver termination is never treated as evidence of infeasibility. Numerical cleanup is checked against the simplex, service and order rows.

A.3 Zero and positive performance cost

Proposition A.1 (When persistence is free)**.** *Under the assumptions of Theorem 3.2, let $\mathcal{O}_k$ be the set of independent minimizers for tier k. Then $\Delta^\star = 0$ if and only if there exist $w_k \in \mathcal{O}_k$ jointly satisfying Eq.* (2)*.*

For any $w_k \in \mathcal{P}_k$, its worst-task fitted risk is at least $r_k^\star$. All tier excesses in Eq. (5) are therefore nonnegative. If independent optimal policies satisfy the joint order, they yield zero maximum excess. Conversely, if $\Delta^\star = 0$, compactness provides an attaining family; nonnegativity forces every tier to attain its independent optimum. That family lies in $\prod_k \mathcal{O}_k$ and satisfies the order. This proves Proposition A.1, including cases with nonunique independent optima.

A.4 How a small performance allowance removes withdrawal

The zero-cost condition extends to near-optimal policies. On the same $\mathcal{P}_k$ as Eq. (4), let $R_k(w)$ denote worst-task fitted risk. For $\eta \geq 0$, define

$$\mathcal{O}_k(\eta) = \{w \in \mathcal{P}_k : R_k(w) \leq r_k^\star + \eta\}, \qquad \rho(\eta) = \min_{W \in \prod_k \mathcal{O}_k(\eta)} D(W), \tag{9}$$

where $D(W) = \max_{(k,\ell)\in E, r}[F_k(t_r) - F_\ell(t_r)]_+$ and E is the set of adjacent tier edges. Thus ρ is the smallest possible acceptance-probability drop after allowing every tier a fitted-risk increase of at most η.

Proposition A.2 (Withdrawal across near-optimal policies)**.** *Under the assumptions of Theorem 3.2, ρ is nonincreasing and $\rho(\eta) = 0$ if and only if $\Delta^\star \leq \eta$. For a fixed allowance, ρ is the optimum of a linear program.*

Proof. Positive denominators make the risks continuous. Each $\mathcal{O}_k(\eta)$ is nonempty and compact, so the continuous function D attains its minimum. Increasing η enlarges every feasible set, hence cannot increase ρ. An attained zero gives an ordered family with all tier excesses at most η; conversely any such family gives zero. This is precisely the defining condition for $\Delta^\star \leq \eta$. To compute ρ, minimize $d \geq 0$ subject to the simplex and service constraints, $[N_e^{\text{fit}} - (r_k^\star + \eta)C_e^{\text{fit}}]w_k \leq 0$, and $F_k(t_r) - F_\ell(t_r) \leq d$ on every adjacent edge and threshold prefix. All rows are linear. □

Unlike measuring one independently fitted family, this diagnostic searches over all equally good or near-optimal choices. A positive $\rho(0)$ rules out resolving the conflict solely by selecting a different independent optimum. Its first zero measures how much fitted performance must be traded for persistent acceptance. It is not a statement about test accuracy.

A.5 Certifying a fitted service family

The deployed family can be certified after fitting without bounding every threshold–rule pair. Fix at most H candidate families, including their score functions, weights, hash algorithms and keys, before using an independent service sample. For a candidate h and tier k, let $a_{hk}(x) \in \{0, 1\}$ be its deployed acceptance and $g_{hk}(x) \in [0, B]$ its total per-arrival resource cost. The bound $B > 0$ must hold for every request, not just observed requests.

Proposition A.3 (Simultaneous service certification)**.** *Suppose the certification requests are i.i.d. from the deployment workload, independently of the fixed candidate families. Use the first n requests*

in their sampling order, or a permutation independent of their values. For S prespecified sample sizes, set $\alpha = \delta/(2HKS)$. At each size $n \geq 2$, define

$$L_{hk} = \bar{a}_{hk} - \sqrt{\frac{\log(1/\alpha)}{2n}}, \tag{10}$$

$$U_{hk} = \bar{g}_{hk} + \sqrt{\frac{2\widehat{v}_{hk}\log(2/\alpha)}{n}} + \frac{7B\log(2/\alpha)}{3(n-1)}, \tag{11}$$

where $\widehat{v}$ is unbiased sample cost variance. With probability at least $1-\delta$, all these bounds hold simultaneously. Consequently any selected family satisfying $L_{hk} \geq \gamma_k$ and $U_{hk} \leq \beta_k$ at every tier meets all population coverage and expected-cost requirements.

Proof. For each fixed family, tier and sample size, Hoeffding's inequality bounds acceptance from below with error probability at most α. The one-sided empirical Bernstein inequality bounds cost from above with the same probability after scaling by B (Maurer & Pontil, 2009). A union bound over the $2HKS$ events gives the result. Selection can depend on these bounds because they hold for the entire candidate collection. Nested samples and different tiers need not be mutually independent. □

Clipping the bounds to the known ranges is valid. The candidate library and thresholds may depend on earlier fitting or calibration data; the certification sample must remain separate. No correctness labels are required. A single family is selected for the entire tier order, preserving Theorem 3.1. Failure to certify is reported explicitly. This result controls expected resource use under the stated workload model, not cumulative expenditure or arbitrary distribution shift. An unbounded archived token charge cannot be certified by treating its observed maximum as B.

A.6 When a fitting improvement transfers to routing quality

Joint fitting protects the most affected tier in the estimated problem. The following condition makes its relation to population quality explicit. Use the same fixed service-feasible sets for both comparisons and suppose $|\widehat{R}_k(w) - R_k(w)| \leq \varepsilon$ uniformly over these sets. Let $r_k^\star$ and $\widehat{r}_k^\star$ be their population and fitted independent optima, and define $J(W) = \max_k[R_k(w_k) - r_k^\star]$ and its fitted counterpart $\widehat{J}$. Then

$$|\widehat{J}(W) - J(W)| \leq 2\varepsilon, \qquad J(\widehat{W}) \leq \min_{W \text{ ordered}} J(W) + 4\varepsilon + \varepsilon_{\text{alg}}, \tag{12}$$

where ε_{alg} bounds the total fitted-objective suboptimality of the returned ordered family $\widehat{W}$. Indeed, uniform risk error bounds each independent optimum's error by ε. Subtracting the optima and taking a maximum gives the first inequality; applying it twice around the fitted optimum gives the second. For any ordered comparator V, $J(V) - J(\widehat{W}) \geq \widehat{J}(V) - \widehat{J}(\widehat{W}) - 4\varepsilon$. Thus a fitting advantage exceeds estimation uncertainty only when the gap is sufficiently large. This is a sufficient condition, not an asserted numerical certificate for our benchmark risks. It motivates the matched risk-estimation ablation and the independent test of tier protection.

A.7 Why an early tier choice can restrict later tiers

The routing rule can change with the budget, but persistent acceptance couples which requests those rules may serve. The following observation identifies the decision that joint fitting coordinates.

Proof of Proposition 3.3.

Proof. The difference is nonnegative. Its expectation equals the coverage difference; if that expectation is zero, it vanishes almost surely. □

Thus, at equal coverage, a tierwise fit commits later tiers to its first acceptance function on the workload support. Changing models alone cannot undo that commitment. At unequal coverage, the same identity measures the available increase in acceptance. This statement concerns the specified distribution; empirical equal coverage establishes equality on the calibration support, not on every unseen score interval.

An exact construction: positive cost and a joint-fitting benefit. Consider two equally likely request types with scores 0 and 1. A cheap model costs 0.1 and has conditional errors $(1, 0.4)$; an expert costs 1 and has errors $(0, 0.5)$. The library contains both constant-model rules and a rule that calls the expert on type 0 and the cheap model on type 1. Thresholds are $-\infty, 0.5, +\infty$. Both tiers have coverage $1/2$, with cost caps 0.05 and 0.5. These numbers define a constructed distribution, not an additional LLM experiment.

Write the common acceptance probabilities as $(p, 1-p)$, where $0 \le p \le 1/2$; threshold weights $(p, 1-2p, p)$ realize them. The low budget permits only cheap accepted calls, giving risk $0.4+0.6p$. At the high tier, the query-dependent rule gives risk $0.4(1-p)$ and costs $0.05+0.45p$. The constant expert is dominated by that rule. Independent optimal risks are therefore 0.4 and 0.2. Tierwise fitting fixes $p = 0$ at the first tier and incurs maximum excess 0.2. Joint fitting instead solves

$$\min_{0\le p\le 1/2} \max\{0.6p,\ 0.2 - 0.4p\} = 0.12, \qquad p = 0.2.$$

Thus persistence has strictly positive minimum cost $\Delta^\star = 0.12$, and joint fitting reduces the tierwise cost from 0.20 to 0.12. The improvement holds with exact risk knowledge, identical coverage and the same model-selection library. Exact rational arithmetic and an independently assembled LP recover these values. More generally, excesses ap and $b(1/2 - p)$ give joint excess $ab/[2(a + b)]$, versus $b/2$ for the tierwise commitment. No benefit is forced when their preferred acceptance functions already agree. Together with Eq. (12), this separates the existence of a coordination gap from the accuracy needed to estimate it.

A.8 Shared prefix calls and exact service matching

Let v contain a fixed shared prefix response, its score and its cost b. For candidate policy $m = (i, j)$, acceptance is $I_m(v) = \mathbf{1}\{s(v) \ge t_i\}$, and its cost is $b + I_m(v)c_{h_j(v)}$. Since $\sum_m w_m = 1$,

$$G(w) = \mathbb{E}[b] + \sum_m w_m \mathbb{E}[I_m c_{h_j}] = \sum_m w_m \mathbb{E}[b + I_m c_{h_j}].$$

Thus nonzero rejection cost preserves linear mixture moments. Finite nonnegative cost moments suffice for the fitting theorem; no concentration guarantee for unbounded charges is asserted. Conditional on the same prefix realization v, the shared-quantile proof still gives $I_k(v, U) \le I_\ell(v, U)$. Integration over prefix randomness preserves the result when that realization is shared across tiers. Independently regenerated responses are outside this comparison.

Adding fixed affine rows, including exact calibration coverage, preserves the compact convex feasible sets. Given positive fitted denominators and joint feasibility, the fitting, zero-cost and near-optimal compatibility results remain valid. For a single task, the fractional program $\min_w Nw/(Cw)$ subject to $Aw \le d$, $Hw = f$, $w \ge 0$ (including $\mathbf{1}^\top w = 1$) has an exact LP: set $t = 1/(Cw)$ and $y = tw$, impose $Cy = 1$, $Ay \le dt$, $Hy = ft$, $y, t \ge 0$, and minimize Ny. The transformed simplex enforces $\mathbf{1}^\top y = t$; $Cy = 1$ implies $t > 0$, so $w = y/t$ proves the converse. This gives an equivalent LP for the same objective.

B Experimental protocol

B.1 Reusable routing interface

Error forecasts $q_a(x)$ supply the shared score and model-selection rules:

$$s(x) = -\min_a q_a(x), \qquad h_j(x) \in \arg\min_a \{q_a(x) + \lambda_j \widehat{c}_a(x)\}, \tag{13}$$

with fixed tie-breaking, $\lambda_j \ge 0$, and additional constant-model rules. The acceptance theorem permits any deterministic routing rules with a shared score.

B.2 Data roles and model pools

The archived LLMRouterBench release used here contains 13 models and ten performance–cost datasets; we use the four tasks in Section 4.1. Stratified hashing of task and record identifiers assigns

fractions $0.25/0.10/0.10/0.10/0.45$ to predictor training, policy fitting, validation, calibration and testing. The six overlapping partitions are splits 1–6; the primary MLP uses 2, 5 and 6. Assignments and randomization settings accompany the code. This corpus was used in development and supports architecture reuse, not independent confirmation.

Roles are record-disjoint, but one normalized prompt occurs in two domains and its copies may cross roles. Bootstrap resampling groups these copies. We do not claim prompt-disjoint generalization here. The smallest test task has 475 records. Identifiers use Unicode NFKC, collapsed whitespace and SHA-256.

We use the release's `cost` fields without repricing (Li et al., 2026a). Training-only mean costs select five models across five cost strata. Selected costs are min–max normalized, $c_a = (\bar{c}_a - \min_b \bar{c}_b)/(\max_b \bar{c}_b - \min_b \bar{c}_b)$. Zero means the cheapest selected model, not a free call. Every method within a split uses these same fixed training-derived costs, not realized token bills or independently verified current API prices.

Table 5: Five-model pools and normalized costs. The first four models, in cost order, are Qwen3-235B-A22B-2507, GPT-5-chat, Qwen3-235B-A22B-thinking-2507 and DeepSeek-R1-0528. The last column gives the fifth model. Each split uses the same pool across its router architectures.

Split	Normalized costs in model order	Fifth model
1	0.000 / 0.114 / 0.233 / 0.453 / 1.000	GPT-5
2	0.000 / 0.108 / 0.221 / 0.445 / 1.000	GPT-5
3	0.000 / 0.053 / 0.103 / 0.218 / 1.000	Gemini 2.5 Pro
4	0.000 / 0.054 / 0.110 / 0.219 / 1.000	Gemini 2.5 Pro
5	0.000 / 0.054 / 0.110 / 0.218 / 1.000	Gemini 2.5 Pro
6	0.000 / 0.055 / 0.111 / 0.223 / 1.000	Gemini 2.5 Pro

B.3 Router adaptations

The primary predictor is a width-96 MLP on frozen `BAAI/bge-base-en-v1.5` embeddings. It uses 1,600 optimizer steps, 64 training examples per task in each step and a smooth worst-task objective with temperature 0.12. No LLM expert weights are updated.

RouteLLM-style utilities use matrix-factorization predictions from the upstream implementation adapted to this model pool. EmbedLLM-style TextMF uses frozen text embeddings, a 128-dimensional projection and model embeddings, and a two-class classifier on their elementwise product. It is trained for 200 epochs on predictor-training correctness labels, with Adam learning rate 10^{-3}, weight decay 10^{-5} and embedding noise 0.05. RouterDC-style utilities use a 128-dimensional query projection and normalized model embeddings, a temperature-0.1 contrastive objective against per-query performance, and a model-profile similarity penalty of weight 0.1. Training uses 300 epochs, batch size 512 and the same Adam settings. This last model is an adaptation of the design, not a faithful reproduction of the complete original RouterDC training pipeline.

Architecture utilities are mapped monotonically to $[0, 1]$ using the minimum and maximum on predictor-training data, clipped, and converted to error proxies $q = 1 - \text{utility}$. They need not be calibrated error probabilities. Predictions and fitting-score threshold grids are fixed across policy methods. Predictor training is excluded from the incremental cost of the policy layer.

B.4 Library, risk moments and selection

The nine tiers cross coverage targets $\{0.65, 0.75, 0.85\}$ with normalized budgets $\{0.20, 0.50, 0.80\}$. All methods use calibration margins $\epsilon_C = \epsilon_G = 0.05$ and the same threshold grid below. The 23 thresholds are 21 fitting-score quantiles plus $-\infty$ and $+\infty$. The full library crosses them with 11 cost-weighted rules and four additional constant rules. The largest cost weight λ_j already chooses the cheapest model, giving all five constant rules. The alternative library retains just those five constants. Its acceptance score is still query-dependent, and its action distribution can depend on the sampled acceptance threshold.

For task e with n_e fitting records, the accepted-error moment is

$$N_{em}^{\text{fit}} = \frac{n_e}{n_e + \tau} \widehat{N}_{em}^{\text{outcome}} + \frac{\tau}{n_e + \tau} \widehat{N}_{em}^{\text{prediction}}, \qquad \tau \in \{16, 64, \infty\}. \tag{14}$$

The observed statistic uses measured errors; the predicted statistic replaces each error with $q_a(x)$. The parameter τ is the effective weight of the predicted statistic, and $\tau = \infty$ uses predictions alone. Coverage remains empirical. Every method independently chooses the library and τ with best mean validation worst-task accuracy. Ties prefer the full library, then the earlier τ in the listed order; tierwise fitting additionally breaks ties by its prespecified order. The constant-rule library is selected by CoRoute in 16 of 21 cases. Test outcomes do not select a margin, library, τ or tier-fitting order.

Controls that preserve acceptance. Tierwise fitting minimizes each tier's worst-task risk under fixed predecessor constraints; validation selects row-first or column-first order. The matched comparisons specify single-cut mixtures and single-model controls in Sections F and G. All methods use the same service requirements, with their actual operating points reported alongside accuracy.

Evaluation and numerical checks. Selective risk divides total wrong mass by total accepted mass, integrating the frozen mixtures over cached outcomes. Macro accuracy averages four tasks; cost and coverage average tasks and tiers. Architecture means weight all 18 adapted cases equally, retaining service misses.

Paired 95% intervals use 2,000 normalized-prompt-group bootstrap draws within each split, shared across methods and tiers. They are pointwise percentile intervals, not independent replication across overlapping splits. Full per-split contrasts and service checks accompany the code. Direct evaluation and mixture statistics agree within 2.3×10^{-16}; independent checks verify fitted LP bounds and comparator objectives.

C Design and performance-cost diagnostics

These analyses use the frozen predictors and policies from the main study. They explain the design on the existing development corpus; no test outcome changes a policy, hyperparameter or deployment choice.

C.1 Why severe withdrawal can have a small performance cost

Using each fit's selected library, risk estimates and service constraints, we minimize Proposition A.2's ρ over all near-optimal policies. At zero allowance, all 21 cases retain conflicts, with minimum probability drops of 6.25–50.51 points. All reach zero by a 0.50-point risk allowance; their exact fitted costs lie between 0.015 and 0.487 points.

Allowances are $\{0, 0.025, 0.05, 0.10, 0.15, 0.25, 0.50\}$ points. Primal/dual checks and independently rebuilt LPs agree with the joint risk bounds.

C.2 Does a larger routing library always help?

Each library uses its validation-selected τ on all 21 cases. The constant-rule library obtains 18.18% mean test worst-task accuracy versus 17.95% for the full library, but uses higher cost (0.442 versus 0.417) and lower coverage (0.848 versus 0.862). These differences do not establish superiority at equal operating points. They show why library capacity is selected on validation rather than assumed to improve test performance. The smaller library is selected in 16 of 21 cases. Its acceptance remains query-dependent: the shared score determines acceptance, and the sampled threshold can change the conditional distribution over models. This is not one fixed model per tier. In the five selected full-library fits, query-dependent model-selection rules account for 0.45–45.89% of accepted probability mass; the equal-case mean across all 21 fits is 6.10%. Thus the shared score is the main query-dependent interface in these experiments, and strong use of query-dependent model selection is not demonstrated in every router adaptation.

The supplement lists the per-case rule weights. The separate 20-model confirmation in Section H tests a setting with substantial query-dependent model selection under the same policy-layer interface.

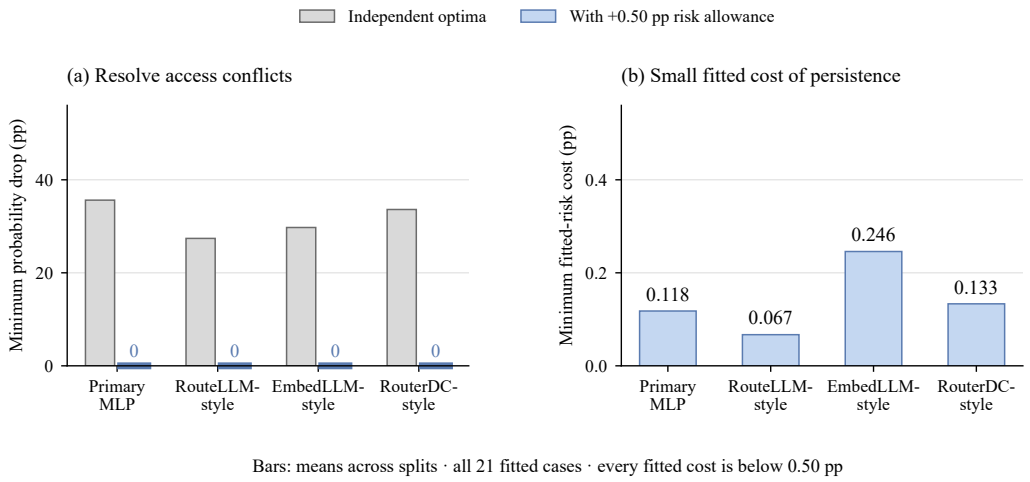

Figure 4: Resolving service conflicts at small fitted cost. Left: minimum acceptance-probability drop after searching all independent optima (gray), versus zero after allowing at most 0.50 points of extra fitted risk per tier (blue). Right: the exact minimum fitted-risk cost of persistence, $\Delta^\star$. Bars are means across splits (three primary, six per adapted architecture). All 21 configurations are included, and every fitted cost is below 0.50 pp. Both axes start at zero. These fixed-library diagnostics explain the fitted optimization problem; they are not test accuracy gains.

D When joint fitting protects a tier

The ablation fixes the routing library, service constraints and deployment, changing only joint versus tierwise fitting. These diagnostics separate coordination gains from risk-estimation uncertainty.

Known-risk diagnostic. Prompt-deduplicated fitting and validation records define seven finite populations: three primary, one per adapted router, and the 20-model population. Their recorded outcomes give exact risk and service moments for fixed predictors and rules. Cuts use 13 score quantiles and two endpoints. Three or five coverage levels spanning 0.60–0.90 cross budgets $\{0.20, 0.50, 0.80\}$ or $\{0.12, 0.24, 0.48\}$, yielding nine or fifteen tiers. Both solvers receive the same moments and exact coverage constraints. Tierwise fitting uses the better of its two legal orders; Independent uses the same population.

Table 6: Known-risk coordination diagnostic. Loss is the largest risk increase over the population Independent optimum (pp), averaged over configurations. These are finite-population optima, not held-out scores. Configurations share requests and are not independent replications.

Population / objective	Configs.	Tierwise loss	Joint loss
Multi-task / worst task	24	1.186	**0.436**
Multi-task / pooled	24	0.402	**0.170**
20 models / pooled	4	0.000	0.000
20 models / worst task	4	0.001	0.001

All 24 multi-task configurations benefit, by 0.16–1.70 points. Pooling risk while fixing the service mixture reduces the gap. In the 20-model population, independent optima are already compatible to fitting tolerance. Workload conflict, rather than tier count or budget tightness alone, determines the gain.

Risk estimation with less labeled data. For nine tiers, four nested task-stratified samples without replacement at each size change only risk estimates; population service moments stay fixed. Both methods use the same samples. Each internal population has 2,107 records and the 20-model pop-

ulation has 1,654; samples are capped and rounded within task. Evaluation uses the entire finite population, including sampled requests, so this diagnoses estimation rather than fresh-test performance.

Table 7: Effect of labeled sample size on the finite-population diagnostic. All 28 context–repetition runs per size are included. Win/tie/loss compares joint with tierwise population loss beyond the numerical tolerance.

	Largest tier loss (pp)		
Sample cap	**Tierwise**	**Joint**	**Win/tie/loss**
128	7.61	7.28	11/5/12
512	3.27	2.47	16/3/9
2,048	1.15	0.50	24/4/0

Larger samples make population gains more consistent. Compatible policies leave little room to improve; uncertain risks can obscure an existing gap.

Additional held-out outcome check. MMLU records from the same response release (Hendrycks et al., 2021; Wikit, 2026) exclude prior full-prompt, question-core and eight-word-sequence matches, plus internal duplicates. The remaining 1,045 records from 48 subjects split into 208 predictor-training, 484 fitting, 138 calibration and 215 test records. Models and predictor match Section H. Official categories define four risk groups. Both worst-domain and pooled objectives use equal-domain service weights; pooled risk divides mean error mass by mean coverage.

Exact calibration coverages are $\{0.60, 0.75, 0.90\}$, nominal targets 0.05 lower. Ordinary budgets $\{0.20, 0.50, 0.80\}$ and tight budgets $\{0.04, 0.08, 0.16\}$ use the same 10% guard for every method. Policies and analyses are frozen before final-test access. Intervals use 4,000 domain-stratified prompt resamples and Bonferroni correction for four contrasts (98.75% each); worst-domain risk with tight budgets is primary. This is a new cohort within an existing benchmark family, not a new-domain or live-service test.

Table 8: Joint versus tierwise fitting on the additional held-out cohort. Positive loss reduction favors joint fitting. Intervals account for four optimizer contrasts. Passes require both nominal coverage and budget on the test sample; they are empirical checks, not certificates.

Risk	**Budget**	**Loss reduction (pp)**	**Interval**	**Joint/tierwise passes**
Worst domain	Ordinary	+3.36	$[-4.57, +9.99]$	6/9; 6/9
Worst domain	Tight	-0.72	$[-5.29, +3.92]$	6/9; 6/9
Pooled	Ordinary	+0.00	$[-1.72, +1.15]$	3/9; 6/9
Pooled	Tight	+0.92	$[-1.75, +4.43]$	3/9; 9/9

The small final cohort gives broad intervals and establishes no stable joint–tierwise advantage. The known-risk diagnostic establishes the coordination mechanism; the held-out comparison measures how reliably the current risk estimator realizes its benefit.

E WITHDRAWAL IN A PUBLISHED ABSTAINING CASCADE

We replay the original Early Abstention policies (Zellinger et al., 2025) on all six tasks and 16 published two-model cascades, without a CoRoute fit. The pools contain Llama 1B/3B/8B/70B/405B and, separately, GPT-4o mini, Qwen 32B Coder, Qwen 72B and GPT-4o, paired as in the original study.

Policies and data. We preserve the released 41×41 preference grid, confidence transformation, training-only logistic calibration, threshold smoothing ($r = 10$), and recursive answer/defer/abstain rule. Prompt hashing splits original test data into development and reserved halves after excluding matches in 101,852 prior research prompts and the authors' training questions. Development counts are 499/506/734/487/244/483 for GSM8K/MedMCQA/MMLU/TriviaQA/TruthfulQA/XSum. The

reserved half is used only in Section F. All cases, upgrade rules and roles are fixed before outcome evaluation.

Upgrade definition and results. An adjacent preference change raises the penalty for abstaining at fixed cost penalty, or lowers the cost penalty at fixed abstention penalty. These are native preference settings, rather than CoRoute's hard service targets. Our specified service-oriented slice retains pairs whose training coverage and mean cost are both nondecreasing and whose coverage lies in $[0.50, 0.95]$ at both ends. Four GSM8K cascades have no eligible pairs. Of 92 eligible task–cascade cases, 79 withdraw requests on at least one selected upgrade. Together they contribute 60,469 adjacent pairs, selected without test outcomes.

Table 9: Withdrawal in the original Early Abstention policies. *Cases* reports cascades with a withdrawal over cascades with eligible adjacent pairs. *Mean* averages the per-edge withdrawal fraction within each case, then across cases; *maximum* is the largest single-edge fraction. Both are percentages of evaluated requests. The grid-wide union is not used as a single-upgrade frequency.

Task	Cases	Mean (%)	Maximum (%)
GSM8K	8/12	0.12	9.42
MedMCQA	16/16	0.15	8.89
MMLU	16/16	0.11	5.99
TriviaQA	14/16	0.16	7.19
TruthfulQA	13/16	0.08	11.07
XSum	12/16	0.02	10.97

The result is not created by threshold smoothing: raw archived thresholds produce withdrawal in 78 of the 92 eligible cases. The authors' final-model-only abstention control also withdraws service in 60 of 86 eligible cases. Full preference-grid and service-ordered summaries are available in the replay tables. Cases share requests within a task and are not independent replications.

Cost and implementation. Every invoked model is charged using the released per-request `model_costs`, including calls before abstention and free-form confidence verification. These archived costs are not current API bills. Vectorized decisions match the pinned recursive implementation on 2,415 synthetic boundary checks and 19,200 real policy–request checks. The supplement provides versions, replay code and full preference-grid summaries.

F Independent external component confirmation

Policies, case inclusion and primary comparisons are frozen before evaluating the reserved corpus from Section E: all 96 task–cascade cases in both coverage settings. Mean accuracy, mean maximum tier loss and the largest observed loss are prespecified. Task breakdowns, additional tail summaries and outcome transitions below are secondary, post-hoc analyses of the same fixed policies, without refitting or changing case inclusion.

Data and score availability. Author training data fit confidence and risk moments: 300 records per task, 285 for MMLU after overlap removal. Prompt hashing divides development data between unlabeled service calibration and labeled validation. Reserved data contain 3,076 records (3,061 distinct prompts), with task counts 501/494/778/513/273/517 in Table 9's order. Duplicates stay in one role and bootstrap group. Each task has its own fitted cascade. We preserve the release's binary success labels. For XSum, success requires 20/20 summed over its four summary ratings: coherence, consistency, fluency and relevance. Thus "correct" below means meeting the archived task criterion, rather than a common factual-correctness test across all six tasks.

Every request first invokes the earlier model. Its calibrated confidence supplies the shared score; acceptance returns its cached answer or invokes the second model. The component cannot observe the second model's confidence or correctness before deciding. Compared tiers reuse the prefix response. The library has nine training-decile deferral rules plus always-first and always-second rules. Cuts use 81 training quantiles, two infinite endpoints and the calibration boundaries needed for exact coverage.

Service targets and complete call costs. Write $\bar{c}_1, \bar{c}_2$ for calibration mean archived call costs and q for the training mean cost of calling both models. For coverage target $\gamma \in \{0.50, 0.65, 0.80\}$ and incremental fraction $f \in \{0.15, 0.35, 0.55\}$, fitting enforces coverage at least $\gamma + 0.05$ and mean cost at most $(\bar{c}_1 + f\bar{c}_2)/q$. The equal-coverage setting replaces the first inequality with equality. Nominal test budgets include a fixed slack $0.05\bar{c}_2/q$, chosen before confirmation. For each arrival, cost is $c_1 + I\,\mathbf{1}\{\text{defer}\}c_2$, including the first call on declines. The full archived second-model charge is included even when its confidence is unused. Thus reported costs include the archive's confidence-verification charges and are not current API prices.

Controls. Independent, CoRoute and tierwise fitting share the library, risk statistics and service constraints. Tierwise fitting chooses between two legal orders on validation. Single-cut mixtures use exact boundary randomization and the same routing rules; validation chooses globally optimized fitted mean risk or maximum excess risk. Small exhaustive problems verify their mixed-integer LPs.

The native cascade retains its published two-model decision rule. Each tier selects calibration-feasible thresholds on validation from the archived smoothed policies, a 21-quantile training grid for each of three thresholds, and pure-first-model policies. Equal-coverage evaluation thins answers uniformly without refunding calls. Union repair returns an accepted answer from the requested tier or its ancestors, in decreasing row-major priority. It charges all required calls, caching each model once; its union can increase coverage and cost.

Fitting and deployment. Independent and tierwise subproblems use Section A.8's exact LP; CoRoute uses 25 cap-bisection steps and a tie-break within 10^{-5} of the cap. All fitted denominators are positive. Independent reconstruction of 768 selected policy arrays gives maximum service residual 1.3×10^{-13}. The deployment adapter agrees with direct inverse-CDF acceptance on 141,504 validation replays and takes 30.9–32.7 microseconds per tier, excluding the first-model computation and LLM calls.

Performance and uncertainty. Accuracy divides summed correct mass by summed accepted mass. We average nine tiers within each case and weight all 96 cases equally. Loss uses Independent in the same setting. Paired intervals use 2,000 task-stratified prompt-cluster resamples shared across model pairs and methods. Mixture moments are exact; withdrawal also uses a shared SHA-256 prompt draw.

The prespecified 0.5-point mean noninferiority allowance passes: one-sided 95% loss bounds are 0.066 and 0.058 points for minimum and equal coverage. This concerns average loss, not noninferiority at every tier. Tables 12 and 13 supplement the main equal-coverage results with minimum-coverage sensitivity and paired contrasts. No policy or case changes after confirmation.

Tier protection across tasks and loss summaries. At equal calibration coverage, mean maximum tier loss is lower than single-cut mixtures on every task: reductions are 0.27 points on GSM8K, 0.52 on MedMCQA, 0.16 on MMLU, 0.80 on TriviaQA, 1.09 on TruthfulQA and 0.58 on XSum. CoRoute improves 74 of 96 cases; omitting any one task leaves a mean reduction between 0.47 and 0.65 points. These descriptive checks use all fixed cases.

For each prompt-cluster resample, we recompute tier accuracies, each case's maximum loss, and the mean of its ten largest case losses (approximately the worst tenth). Table 10 reports CoRoute minus comparator; negative values favor CoRoute. The single-cut comparison supports protection beyond one extreme; all tierwise intervals include zero. These additional pointwise intervals are secondary analyses conditional on fixed policies and tasks; they do not include fitting or task-population uncertainty.

What service is preserved? We integrate shared threshold and routing draws to identify adjacent upgrades where Independent accepts then declines, while CoRoute accepts at both tiers. At equal calibration coverage, 70.8% of these retained higher-tier answers are correct (paired interval $[69.4, 72.2]\%$); 29.2% are incorrect. Independent's lower-tier answers were correct on 62.2% of the same events. Per 1,000 arrivals and adjacent upgrades, case averages are 3.09 correct and 1.27 incorrect retained answers.

Table 10: Paired differences in largest-tier-loss summaries (pp), with 95% intervals. All cases are included; the tail is recomputed within each resample.

Coverage setting	Loss summary	vs. Single-cut	vs. Tierwise fit
Minimum	Mean	-0.80 [-1.09, -0.79]	-0.03 [-0.14, +0.05]
Minimum	90th percentile	-1.52 [-2.48, -1.37]	-0.10 [-0.52, +0.21]
Minimum	Worst-ten mean	-2.53 [-3.41, -1.99]	-0.54 [-1.15, +0.04]
Equal	Mean	-0.57 [-0.90, -0.54]	-0.00 [-0.12, +0.07]
Equal	90th percentile	-1.30 [-2.51, -1.10]	+0.07 [-0.50, +0.32]
Equal	Worst-ten mean	-2.33 [-3.56, -1.81]	-0.43 [-1.03, +0.16]

This subset covers 53.9% of Independent's withdrawn event mass. Refitting changes lower-tier acceptance, so CoRoute preserves its own accepted requests, not every request served by Independent. These are edge events, not a union over upgrades. Minimum coverage gives 70.7% correct retained answers.

Correctness across upgrades. Table 11 measures outcome transitions over all arrivals, averaged over adjacent upgrades and cases, integrating shared threshold and rule draws against archived labels. Correctness regressions remain. Relative to tierwise fitting, CoRoute's correct-to-incorrect rate is 0.017 points higher (paired 95% interval $[0.009, 0.025]$). Relative to single-cut mixtures, it is 0.082 points lower ($[0.066, 0.098]$). Persistent service and answer-quality monotonicity must therefore be evaluated separately.

Table 11: Correctness transitions per adjacent upgrade under equal calibration coverage (% of arrivals). These secondary fixed-policy diagnostics use all 96 cases and the same paired prompt-cluster resampling as the tail analysis.

Method	Correct to decline	Correct to incorrect	Incorrect to correct
Independent	0.513	1.309	1.935
Single-cut mixtures	0.000	1.473	2.048
CoRoute (tierwise fit)	0.000	1.374	2.072
CoRoute	0.000	1.391	2.100

Where do service constraints fail? At equal coverage, 42 of 77 budget misses occur on XSum and 15 on TruthfulQA. The largest budget excess is 0.0481 normalized units (9.74% relative). The nine coverage misses occur on MMLU (six) and TruthfulQA (three), with maximum shortfall 0.379 points. Every missed point remains in the performance summaries; the complete case–tier breakdown accompanies the results. These fixed-margin results are empirical service checks.

Table 12: CoRoute minus each comparator in mean accuracy (pp), with paired 95% intervals from 2,000 task-stratified prompt-cluster resamples. Prompts use the same bootstrap weight across model pairs and settings. The largest-tier-loss column of Table 3 addresses a different performance objective.

Comparator	Minimum coverage	Equal coverage
Independent	-0.048 [-0.068, -0.028]	-0.039 [-0.062, -0.018]
CoRoute (tierwise fit)	-0.026 [-0.059, +0.006]	-0.026 [-0.058, +0.006]
Single-cut mixtures	+0.123 [+0.032, +0.213]	-0.031 [-0.140, +0.076]
Native cascade	+0.554 [+0.332, +0.775]	+0.556 [+0.333, +0.778]
Union repair	+1.288 [+1.076, +1.508]	+1.296 [+1.081, +1.514]

Table 13: Minimum-coverage sensitivity of the external confirmation. Equal-coverage results are in Table 3. All 96 cases contribute; pass counts require both nominal coverage and budget out of 864 tier–case points. Separate coverage and budget checks accompany the code.

Method	Accuracy (%)	Cost	Withdrawal (%)	Pass
CoRoute	76.30	0.452	0.00	774
Independent	76.34	0.452	7.93	760
CoRoute (tierwise fit)	76.32	0.451	0.00	755
Single-cut mixtures	76.17	0.448	0.00	755
Native cascade	75.74	0.433	16.57	806
Union repair	75.01	0.527	0.00	421

G Matched routing and service evaluation

All 21 internal router–split cases and 96 external task–cascade cases are included. This follow-up reuses previously evaluated test portions; Section H supplies separate confirmation. Selection uses only fitting, calibration and validation roles.

Equal calibration coverage. Internal tiers cross exact calibration acceptance $\{0.70, 0.80, 0.90\}$ (nominal targets 0.05 lower) with budgets $\{0.20, 0.50, 0.80\}$. Service constraints average tasks equally while risk protects the worst task. All methods share the full 15-rule library and score cuts augmented by exact coverage boundaries. External targets, budgets, full call costs and the 11-rule library follow Section F.

Risk estimation and selection. We compare empirical accepted-error statistics with forecast-smoothed statistics. For the latter, three prompt-group folds calibrate each model's error forecast by one-dimensional isotonic regression; predictions are out of fold on policy-fitting data. For n_e fitting records, the predicted statistic receives weight $64/(n_e + 64)$ and observed error receives the remaining weight. This changes risk estimates, not the shared acceptance score, routing rules or deployed predictor. Both estimators are applied to every optimizer, keeping the risk-estimation comparison separate from the joint-versus-tierwise ablation.

Internal budget margins are $\{0, 0.025, 0.05\}$ normalized units. External margins subtract $\{0, 0.05, 0.10\}$ times mean calibration second-call cost from the original fit budget. Nominal evaluation budgets never change. Half the original service records, grouped by normalized prompt, fit service constraints. Validation first minimizes the largest nominal service violation, then the largest tier risk increase against the common empirical zero-margin Independent reference, then mean risk. Each method uses this same selection rule. Tierwise fitting chooses between row-first and column-first orders before margin selection. Within 10^{-5} of the optimal fitted risk, independent, joint and tierwise fits use minimum mean accepted-error as a fixed tie-breaker. Multi-task risks use 22 bisection steps; a single pooled risk uses an exact fractional-to-linear transformation. Every risk group has fitting coverage at least 10^{-6}. The single-cut schedule is globally optimized; the single-model control restricts its action rule to one constant model per tier. Missing feasible policies, if any, are recorded rather than replaced by another method.

Test aggregation. Internal accuracy protects the worst task; external accuracy is task-specific. All case means receive equal weight. Maximum tier loss refers to the empirical zero-margin Independent policy within each case. Paired intervals use 2,000 prompt-group resamples shared across methods and overlapping cases: a Poisson cluster bootstrap internally and fixed-size cluster bootstrap externally. Complete operating points and all service checks accompany the code.

Table 14: Joint fitting versus tierwise fitting with validation-selected budget margins and identical risk estimators. Paired 95% intervals are in points. Positive score change or loss reduction favors joint fitting; intervals spanning zero do not establish an advantage. All cases with both methods are included.

Corpus	Risk	Accuracy change	Max. loss reduction
Internal	Empirical	$[-0.076, +0.117]$	$[-0.172, +0.224]$
Internal	Smoothed	$[-0.007, +0.185]$	$[-0.012, +0.420]$
External	Empirical	$[+0.104, +0.223]$	$[+0.194, +0.425]$
External	Smoothed	$[+0.004, +0.102]$	$[+0.031, +0.245]$

The external empirical-risk comparison favors the complete validation-selected joint pipeline: accuracy increases by 0.163 points and maximum tier loss falls by 0.295 points. To isolate the optimizer, we also compare joint and tierwise fitting at each identical budget margin. In all six configurations per corpus (two risk estimators and three margins), the paired intervals for both metrics span zero. The selected-pipeline benefit therefore does not establish a stable benefit from joint optimization alone. Smoothing does not consistently improve the component. The selected empirical joint families pass 179/189 internal and 739/864 external nominal test service points. Full method operating points accompany the code.

Why archived call costs exceed a calibration budget. For a fixed two-model policy, expected resource use is $\mu_1 + p_2\mu_{2|\text{call}}$: the first-call mean plus second-call frequency times its conditional mean cost. Its calibration-to-test change decomposes exactly as

$$\Delta G = (\mu_{1,t} - \mu_{1,c}) + (p_{2,t} - p_{2,c})\mu_{2,c} + p_{2,t}(\mu_{2,t} - \mu_{2,c}). \tag{15}$$

This is an accounting identity, not causal attribution. Replaying the original external CoRoute policies recovers all 77 budget misses. Of these, 42 occur on XSum, where both prefix-cost and conditional second-call-cost changes matter. On TriviaQA, increased second-call frequency dominates the mean drift of the eight failed points. The decomposition and every case–tier record are supplied. Validation-selected margins can reduce empirical misses, but archived variable charges lack a verified deterministic bound for a population certificate.

H Independent service-certified model-pool study

Data and fixed router. The 8,038 prompts come from ASDiv, LogiQA, SocialIQA, MathQA and PIQA in the versioned RoutingCompendium EmbedLLM release (Zhuang et al., 2025; Wikit, 2026). We exclude full-string and punctuation-insensitive question-core matches in prior corpora, and two public preview records, before assigning roles. Released fractional scores are preserved.

We sort the 108 models by released resource weight and name, take 20 evenly spaced indices, and normalize by the largest selected weight. These bounded weights define the resource proxy, not provider prices or independently verified parameter counts. Release revisions, models and checksums accompany the code.

Prompt-disjoint roles cover predictor training, policy fitting, calibration, validation, certification and testing. A multi-output ridge predictor ($\alpha = 1$, with intercept) maps normalized released embeddings to clipped error forecasts. It stays fixed across methods. The score is minus the smallest forecast error. Rules minimize forecast error plus resource cost at weights $\{0, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 100\}$, with all 20 constant rules included. Cuts use 41 fitting-score quantiles and service boundaries. Risk is pooled over this workload; Table 15 also reports each task.

Service certification and selection. Nominal coverage levels $\{0.50, 0.65, 0.80\}$ cross resource caps $\{0.20, 0.50, 0.80\}$. All methods fit exact calibration coverage 0.10 above the nominal level and margins $\{0, 0.05, 0.10\}$ below the budget. Both risk estimators and all five methods use these same choices. Independent certification uses 1,994 requests; final testing uses 1,186 other requests. Threshold and rule draws use fixed SHA-256 streams with prefixes `reliable-u-v1` and `reliable-v-v1`. Certification evaluates these deterministic policies. The hash algorithm and prefixes belong to the certified policy version; replacing them requires recertification.

Full-sample certification uses Proposition A.3 with $H = 30$, $K = 9$, $B = 1$ and $\delta = 0.05$. We retain $\alpha = \delta/(2HKS)$ with $S = 5$ for the prespecified checks at 128, 256, 512, 1,024 and 1,994 requests. Formal certification and family selection use all 1,994 requests. Smaller subsets, formed by sorting a separate fixed prompt hash, are sample-size diagnostics: their input-dependent selection does not itself establish the i.i.d. condition for a population certificate at each size. Among families passing all tiers at full size, the frozen validation ranking selects one whole family. If none passes, the method is reported uncertified; no budget or target is relaxed. Every method has a passing configuration at full size. The same-workload i.i.d. assumption is essential; the result is not a guarantee under arbitrary deployment shifts.

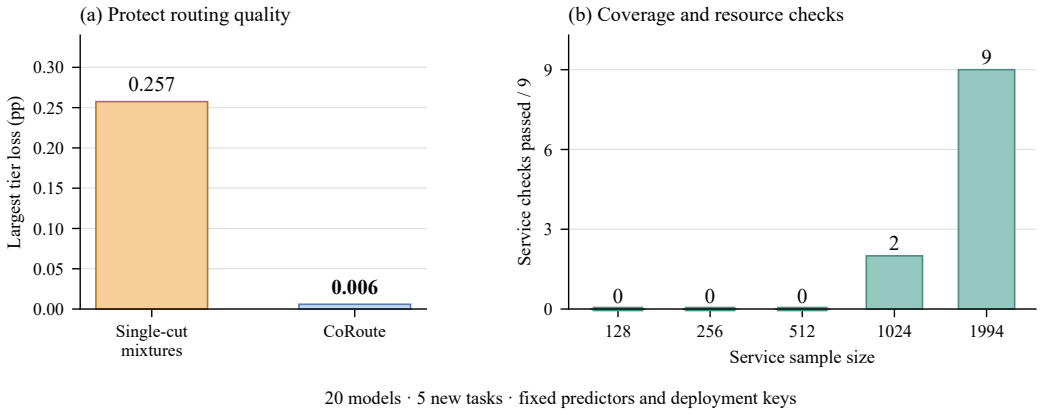

Figure 5: Routing quality and service certification on the new model pool. Left: maximum tier loss relative to certified Independent for the empirical-risk policies; the paired interval for the reduction is $[0.10, 0.92]$ pp. Right: service-bound pass counts for one fixed CoRoute policy and unchanged requirements. Smaller hash-selected subsets provide a sensitivity diagnostic; formal certification uses all 1,994 requests and accounts for every candidate family. Bars show counts, not uncertainty intervals.

Model selection and deployed quality. All accepted mass uses query-dependent routing rules; 16 of 20 models each receive more than 1% of that mass. Table 4 reports quality, coverage and cost, including the lower cost of the single-model control. These comparisons share budget limits, not actual spend.

Fixed-hash deployment scores are 60.189% for CoRoute and 60.188% for Independent, close to their mixture scores in Table 4. Independent withdraws 0.084% of requests; all persistent methods have zero withdrawal and pass all nine final-test service points. The joint–tierwise score difference is 0.004 points with interval $[-0.004, 0.013]$; maximum-tier-loss reduction has interval $[-0.011, 0.039]$. Smoothing changes CoRoute's mean score by 0.012 points with interval $[-0.012, 0.037]$. These results support the complete policy layer but do not establish an additional gain from smoothing or joint fitting on this workload.

Table 15: Per-task mixture scores on the independent final test (%). Each row uses the same frozen empirical-risk family.

Task	Independent	CoRoute	Tierwise	Single-cut	Single model
ASDiv	76.50	76.50	76.49	76.69	76.61
LogiQA	42.31	42.31	42.31	42.03	16.28
SocialIQA	34.25	34.26	34.26	34.74	30.46
MathQA	43.41	43.41	43.40	43.12	25.31
PIQA	81.29	81.29	81.29	81.30	72.07

I ADDITIONAL RESULTS

Independent's shared-draw withdrawal rates span 1.9–28.6% across splits. Architecture means appear in Table 1; the table below gives per-task quality. Maximum tier loss is $\max_k(a_k^{\mathrm{ref}} - a_k)$ within a case, in points, where the reference is Independent. The mean of these case maxima and their largest observed value are distinct from the fitted optimum $\Delta^\star$.

Table 16: Test selective accuracy (%) by task, averaged over the 18 adapted-router cases and nine tiers. HLE determines worst-task accuracy, while macro accuracy weights all four tasks equally. The comparison measures each router before and after adding the component.

Task	Independent	CoRoute
HLE	18.35	18.23
LiveCodeBench	77.35	77.34
MMLU-Pro	85.09	85.05
SimpleQA	44.58	44.96

Service requirements. All router-reuse policies satisfy nominal test budgets. CoRoute's three coverage misses are on LiveCodeBench in RouterDC-style split 5, at the lowest coverage target and all three budgets: 64.77% versus 65%. Independent misses the same points, by at most 0.567 points. Both families and every evaluated point remain in the means. Comparative controls are evaluated at matched calibration coverage in Table 2.

J DEPLOYMENT AND THE REAL FIGURE EXAMPLE

Interface and guarantee. `PolicyLayer` stores thresholds and weights and samples their CDFs using Algorithm 1. Compared tiers share score computation, identifier, key and policy version; model or score-version changes fall outside the contract. The default adapter uses domain-separated BLAKE2b streams for $U, V \in [0, 1)$ and the algorithm's strict CDF boundary convention. Normalized prompt identifiers provide neither paraphrase invariance nor protection against retries with new IDs.

Ideal input-independent uniforms yield the fitted mixture moments. Fixed hashes retain acceptance order under exact CDF constraints, but their workload quality need not equal mixture quality. A service certificate evaluates the actual fixed-hash policy; changing its hash algorithm or key requires recertification (Section H).

Numerical order. The loader's 10^{-10} order tolerance does not imply exact all-draw monotonicity. Positive CDF residuals are below 1.2×10^{-16} in the 21 router-reuse policies and 1.8×10^{-14} across the 24 persistent certification candidates; observed test withdrawal is zero. For pairwise residual η, ideal-uniform withdrawal probability is at most η, which does not certify every fixed request hash.

Timing. The adapter matches all 99,645 sampler replays. Across three primary splits and one per adapted architecture, it takes 32.6–36.1 microseconds per tier and 82.7–91.6 for all nine, excluding embeddings, prediction, network and LLM inference. With moments prepared, independent optima, joint fitting and the L1 tie-break take 1.77–4.33 seconds; full comparison searches take 110.3–235.8 seconds on an Intel Core i9-14900K CPU.

J.1 REQUEST EXAMPLE IN FIGURES 1 AND 2

SimpleQA record 1447 (primary split 5, Q3 in Figure 1) uses 75% coverage and budgets 0.50/0.80, with score -0.0304 and shared draws $U = 0.7169$, $V = 0.0604$. Independent acceptance probabilities fall from 0.8666 to 0.7154: the lower tier calls DeepSeek-R1 and the higher tier declines. CoRoute samples threshold -1 at both tiers, accepting and selecting DeepSeek-R1 and Gemini 2.5 Pro, respectively. Both archived answers are correct, as are the other three models' answers. Forecast errors of 0.9915 and 0.0505 for the chosen models are predictions, not observed error labels. Model logos imply no affiliation or endorsement.