ROUTE ANYTHING: QUALITY–COST TRADEOFFS IN MODEL ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

An inference router chooses which model, tool, or combination should handle a task using the current state and records from earlier outcomes. We study three coupled decisions: what capabilities to combine, when to spend more computation, and what information to retain. We use a reference-relative accounting that compares delivered quality with total resource cost, including routing, verification, and information costs, and apply fractional programming to finite-menu routing with a quality floor. On CodeRouterBench and RouterBench, the accounting shows that selection changes the tasks left to a fallback: comparing both portfolios on the same tasks, instead of freezing the current portfolio's conditional outcomes, raises best-candidate selection on CodeRouterBench from 86.3–94.3% to 96.4–97.1%. With a reference policy and an allowed quality drop, maximizing quality per unit cost amounts to choosing a particular balance between quality and cost. The resulting routers remain effective when routing states are inferred from prompts and perform comparably to learned quality–cost routers. With feedback and scoring tests kept disjoint, feedback retries recover 13.9% more failures than fresh retries (0.304 vs. 0.165) across seven models.

1 INTRODUCTION

An inference router chooses the next computation from a heterogeneous capability pool. It may dispatch to a small model, combine proposals, or switch models after tool feedback. The decision can occur once per task or repeatedly during execution. Its value depends on the final answer delivered and the resources used to obtain it. The central problem is how to choose sufficient capability at a resource cost justified by the value it delivers.

Model routing and cascades formalize parts of this problem. FrugalGPT learns cheaper cascades (Chen et al., 2024b), RouteLLM learns routing from preferences (Ong et al., 2025), and unified formulations connect routing and cascading under quality-cost objectives (Dekoninck et al., 2025). However, different routing strategies show distinct patterns: direct dispatch commits routing decisions before observing an answer; a verified cascade can escalate afterward; and a portfolio additionally chooses which capabilities to use together. These decisions change both the task population reaching later stages and the information available to future decisions. Their value therefore depends on the complete policy, including the cost of acquiring and retaining that information.

To unify the analytical framework, we study inference routers via a unified stateful routing model, comparing delivered quality with total resource cost. We ask:

> What should a router combine, when should it spend more, and what should it remember?

These questions are coupled. Selection-based portfolios gain from newly covered failures; synthesis-based workflows must improve the final answer. Later attempts must be evaluated on the tasks that reach them. A verifier or routing signal must improve decisions enough to justify its cost. A stored record must improve on the prior policy after accounting for exploration, recording errors, and retention costs. Figure 1 illustrates these comparisons.

Our contributions are threefold: **First,** we state one reference-relative accounting for routing, cascades, portfolios, verifiers, signals, and records, and apply fractional programming to finite-menu routing

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.

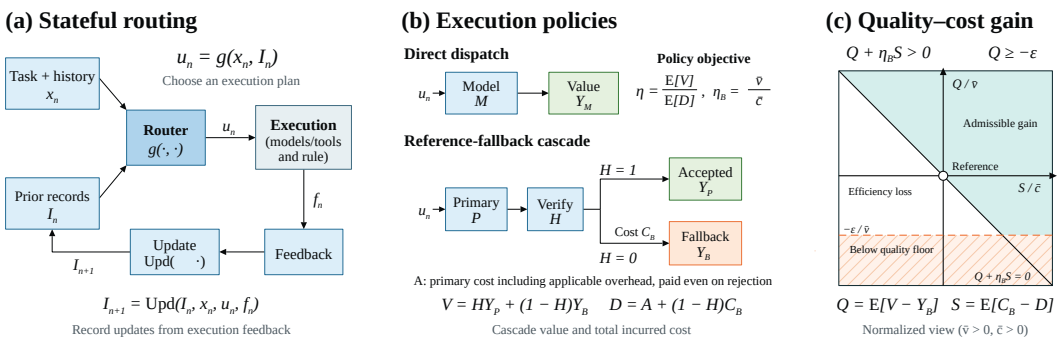


Figure 1: **Stateful routing and its objective.** (a) Feedback updates stored records. (b) Direct dispatch and reference-fallback execution. (c) Admissible gains satisfy $Q + \eta_B \mathcal{S} > 0$ and $Q \geq -\epsilon$.

with a quality floor (Theorem 2); a named reference and tolerance fix the resource price $\lambda = \eta_\epsilon^\star/(1+\kappa)$, the price path yields a finite-sample certified floor (Proposition 3), and the reference both sets the price and competes as an action at that price (Proposition 4). **Second,** we give the exact correction that separates paired portfolio comparisons from residual coverage (Eq. (25)) and measure how often fallback populations shift. **Third,** we test these comparisons through model routing on CodeRouterBench (Zhou et al., 2026) and RouterBench (Hu et al., 2024), supplemented by online retries and semi-synthetic memory streams, which makes values, costs, and failure populations observable on matched tasks. Specifically, we evaluate them on held-out tasks, prompt-inferred states, and learned-router baselines.

2 A Unified Model of Inference Routers

2.1 Policies and Objectives

Router policies and actions. At decision round n, the observable state x_n contains the task and available execution history, while I_n contains records from previous rounds. A stateful router chooses an execution plan and updates its information from feedback f_n:

$$u_n = g(x_n, I_n), \qquad I_{n+1} = \mathrm{Upd}(I_n, x_n, u_n, f_n). \tag{1}$$

The policy may randomize. An action specifies models or tools and an execution rule; the evaluation unit is the completed task, including all continuation costs.

Value, resources, and quality. Let V be delivered task value and D the total consumption of one specified resource, including model calls, routing, aggregation, verification, and information updates. Let Y_B and C_B be the reference policy's performance and associated cost, respectively, evaluated on the same task set. These variables are nonnegative and integrable, with $\mathbb{E}[D] > 0$ and $\bar{c} = \mathbb{E}[C_B] > 0$. Write $\bar{v} = \mathbb{E}[Y_B]$. We compare aggregate efficiencies

$$\eta = \frac{\mathbb{E}[V]}{\mathbb{E}[D]}, \qquad \eta_B = \frac{\bar{v}}{\bar{c}}, \qquad \mathbb{E}[V] \geq \bar{v} - \epsilon, \quad \epsilon \geq 0. \tag{2}$$

We report quality and cost jointly; a ratio gain does not establish quality preservation. At a fixed resource price λ, the objective is $J_\lambda = \mathbb{E}[V] - \lambda \mathbb{E}[D]$.

Resources are compared separately, or combined through stated prices in $\mathbb{E}[V] - \sum_r \lambda_r \mathbb{E}[D_r]$. The benchmark's zero-credit outcome defines the value origin. Shifting it changes ratio rankings, whereas fixed-price differences and the quality floor are invariant (Appendix G.3).

2.2 Reference-Relative Gain and Optimal Routing

Define quality change and resource saving by

$$Q = \mathbb{E}[V - Y_B], \qquad \mathcal{S} = \mathbb{E}[C_B - D]. \tag{3}$$

Lemma 1 (Reference-relative gain)**.** *Under the model above,*

$$\eta - \eta_B = \frac{Q + \eta_B \mathcal{S}}{\mathbb{E}[D]} = \frac{Q + \eta_B \mathcal{S}}{\bar{c} - \mathcal{S}}. \tag{4}$$

Efficiency improves if and only if $Q + \eta_B \mathcal{S} > 0$; the quality floor is the separate condition $Q \geq -\epsilon$.

The identity compares complete policies even when accepted and fallback populations change. More generally, $\Delta J_\lambda = Q + \lambda \mathcal{S}$ prices the tradeoff at an externally chosen λ. For a fixed state distribution, actions can include complete portfolios or continuation plans.

Theorem 2 (Efficiency-optimal routing with a quality floor)**.** *Let the state x take finite values with router-independent probabilities $\pi(x) > 0$. Each state has a finite action menu $\mathcal{U}(x)$ with expected value $v(x, u)$ and cost $c(x, u) > 0$, and a router chooses actions with probabilities $g(u \mid x)$. Put*

$$\Phi(\lambda) = \sum_x \pi(x) \max_{u \in \mathcal{U}(x)} [v(x, u) - \lambda c(x, u)].$$

(i) Φ is convex and strictly decreasing, with unique root $\eta^\star = \max_g \eta(g)$. A router is optimal if and only if it uses only actions maximizing $v(x, u) - \eta^\star c(x, u)$ in every state.

(ii) The iteration $\lambda_0 = \eta_B$, $\lambda_{t+1} = \eta(g_t)$, with g_t a deterministic maximizer of J_{λ_t}, reaches $\eta^\star$ in finitely many steps, monotonically if the reference policy is admissible. Its first step satisfies $\Phi(\eta_B) = \max_g [Q_g + \eta_B \mathcal{S}_g]$.

(iii) If the reference is admissible, then under $\mathbb{E}_g[V] \geq \bar{v} - \epsilon$, an optimal router exists that randomizes in at most one state. A multiplier $\kappa \geq 0$ exists such that every optimal router uses only actions maximizing $(1 + \kappa)v(x, u) - \eta^\star_\epsilon c(x, u)$, where $\eta^\star_\epsilon$ is the constrained optimum.

Here $\eta(g)$, Q_g, and $\mathcal{S}_g$ are the efficiency, quality change, and saving of router g; the reference is *admissible* when it is itself one of these routers. This characterization uses Dinkelbach's method and the Charnes-Cooper transformation (Dinkelbach, 1967; Charnes & Cooper, 1962). Sparse randomization follows from a finite linear program with one quality constraint, consistent with classical constrained-control results (Beutler & Ross, 1985; Altman, 1999). Relative to the routing-cascading optimum of Dekoninck et al. (2025), which maximizes estimated quality under a cost budget, Theorem 2 optimizes the ratio η from the reference, replaces the budget by a quality floor, and bounds randomization to one state. When the floor binds, part (iii) makes the constrained optimum a J_λ optimum at $\lambda_\epsilon = \eta^\star_\epsilon / (1 + \kappa)$, hence also the budget optimum at its own cost, so the ratio adds no new frontier point; it lets the reference and tolerance, rather than a chosen price or budget, select the operating point. The proof is in Appendix H.3. The theorem optimizes routers for a given state law; policy-dependent state transitions are treated separately. Within a task, complete continuation plans from the initial state are the actions (Section 4.1); across tasks, where a router's own writes change later records, Section 5 evaluates a given writer, and optimizing the writer jointly is a constrained Markov decision problem (Altman, 1999).

Proposition 3 (Certified quality floor along the price path)**.** *In the setting of Theorem 2, let values lie in $[0, 1]$ and the reference be admissible with positive fitted value. Fit $\hat{\pi}, \hat{v}, \hat{c}$ on a sample $\mathcal{A}$, and let $g_1, \ldots, g_K$ be the maximizers of the fitted $\hat{J}_\lambda$, with a fixed tie-break, on the successive breakpoint intervals of $\lambda \in [0, \hat{\eta}^\star]$. (a) $K \leq 1 + \sum_x (|\mathcal{U}(x)| - 1)$; along k, fitted value and cost strictly decrease, fitted efficiency strictly increases, and every fitted floor problem is solved by some g_k or a mixture of neighbours. (b) Fix $\epsilon \geq 0$ and let $\hat{Q}_k$ be the mean of $V - Y_B$ under g_k on n tasks drawn i.i.d. from the deployment law, independently of $\mathcal{A}$. With $p_k = \exp\{-n(\hat{Q}_k + \epsilon)_+^2 / 2\}$ and $\hat{k} = \max\{k : p_1, \ldots, p_k \leq \alpha\}$ (the reference if $p_1 > \alpha$), $\mathbb{P}(Q_{g_{\hat{k}}} < -\epsilon) \leq \alpha$. The same holds for any valid p_k and any finite family and order fixed by $\mathcal{A}$.*

The price path fixes the testing order, so no multiplicity correction is needed, as in fixed-sequence testing (Westfall & Krishen, 2001; Angelopoulos et al., 2025); returning the reference when nothing is certified mirrors high-confidence policy improvement (Thomas et al., 2015) (Appendix K.1). The certified interval also yields a *certified price*: reference, tolerance, and level fix λ.

2.3 Model-Routing Specialization

We specialize the action menu to callable language models. Direct dispatch commits to one output. A reference-fallback cascade accepts primary value Y_P when $H = 1$ and invokes the reference

otherwise. With primary-plus-verification cost A, it delivers

$$V = HY_P + (1-H)Y_B, \qquad D = A + (1-H)C_B. \tag{5}$$

A failed primary is still paid for; measured or explicitly priced verification and routing enter A.

Proposition 4 (The reference as yardstick and as input)**.** *In the setting of Theorem 2(iii) with values in* $[0,1]$ *and a reference-free action in every state, let* u *invoke the reference with positive probability, on an event* R_u *(a router is* reference-free *if no action it plays does), and let* $F_x(\lambda)$ *be the largest* $v(x,\cdot) - \lambda c(x,\cdot)$ *of a reference-free action. (a)* u *maximizes* $v(x,\cdot) - \lambda c(x,\cdot)$ *only for* λ *in the closed, possibly empty interval* $\Lambda_x(u) = \{\lambda \geq 0 : v(x,u) - \lambda c(x,u) \geq F_x(\lambda)\}$*, with supremum* $\lambda_x^{\mathrm{hi}}(u)$*. For a cascade* (5) *(*$R_u = \{H = 0\}$*) with any verifier,* $A \geq C_P$ *(the primary's own cost), and direct dispatch to* P *on the menu, a nonempty band has* $\lambda_x^{\mathrm{hi}}(u) \leq \mu_x(u) = \mathbb{E}[(Y_B - Y_P)\mathbf{1}_{R_u} \mid x]/\mathbb{E}[C_B \mathbf{1}_{R_u} \mid x] \leq \mathbb{E}[Y_B \mid R_u, x]/\mathbb{E}[C_B \mid R_u, x]$*. (b) Theorem 2(iii) extends to every real* ϵ *at which some router meets the floor (*$\epsilon < 0$ *demands a gain). Let the* admission price λ^B *be the largest of* 0 *and the* $\lambda_x^{\mathrm{hi}}(u)$ *of nonempty bands, suppose* $\lambda^B < \eta^\star$*, let* $\delta^B = Q_{g^B}$ *for a* J_λ*-maximizer* g^B *at* λ *just above* λ^B *(reference-free;* Q_{g^B} *does not depend on ties), and let* $\delta^{\mathrm{free}} \geq \delta^B$ *be the largest* Q *of a reference-free router. The ratio optimum, and for* $\epsilon > -\delta^B$ *every floor optimum, is reference-free; for* $\epsilon < -\delta^{\mathrm{free}}$ *every router meeting the floor calls the reference.*

The reference thus sets the comparison price and, at that price, competes as an action whose fallback dollar earns at most what the reference earns on the tasks it receives (Appendix H.4).

For a verifier score Σ, define

$$H_\theta = \mathbf{1}\{\Sigma \geq \theta\}, \qquad p_\theta = \mathbb{P}(H_\theta = 1). \tag{6}$$

In replay, Σ is the released score: $\theta = 1$ requires full credit and $\theta = 0.5$ permits partial credit. For a judge, Σ is its rating, not a correctness probability. Since $V - Y_B = H(Y_P - Y_B)$, a cascade loses quality only through false accepts, on any task distribution, whereas a floor for ex-ante dispatch holds only for the law it was fitted and certified on (Proposition 21, Appendix K).

3 What Should a Router Combine?

Data and evaluation. CodeRouterBench (Zhou et al., 2026) records outcomes from eight models on 9,999 coding tasks: 7,080 probing tasks and a 2,919-task test set, partitioned into nine dimensions or 21 dimension×source cells. RouterBench (Hu et al., 2024) supplies eleven models on 36,497 mostly non-code prompts. Candidate selection pools ten stratified splits, each with 25,542 fitting and 10,955 held-out prompts. The reference models are Opus and gpt-4-1106-preview, respectively.

Evidence types. *Replay* results (Sections 3.1–3.2, the chains of Section 4.1, and Test 1 of Section 4.3) use recorded outcomes, with released-score acceptance at $\theta \in \{1, 0.5\}$ as a free oracle verifier and router and verifier fees excluded; they isolate the decision rule. *Online* results (retries, verifiers, and the paid forecast) add fresh calls and reuse recorded Opus outcomes; the memory study is *semi-synthetic*. Measured router and sandbox overheads stay below 0.1% and 3% of a Qwen3-Max call, whereas LLM judges and forecasts cost more than the call (Appendices A.1 and A.9). Intervals use paired task bootstraps; a *decided sign* excludes zero.

3.1 Which Routing Policy Does the Objective Select?

Decision rule and test. We fit action values, costs, and state frequencies on the 7,080 probing tasks and evaluate on the CodeRouterBench test set. States are the 21 non-empty dimension×source cells, or nine dimensions. Each state has 22 actions: direct dispatch to one of eight models, or one of seven non-Opus primaries followed by Opus when its released score is below $\theta \in \{1, 0.5\}$ (oracle replay; token-price costs). Floors $\mathbb{E}[V] \geq \bar{v} - \epsilon$ are imposed on probing for $\epsilon \in \{0, 0.005, 0.01, 0.02, 0.05\}$. Baselines are the best fixed dispatch and cascade on probing, ACRouter's released model map (Zhou et al., 2026), and a per-task in-sample oracle. Always-Opus scores 0.4383 at 11.655 m$ per task; Figure 2 uses 1,000 paired task bootstraps with fitted policies fixed. We also resample probing and test tasks and refit the policy in 500 replicates to include selection uncertainty.

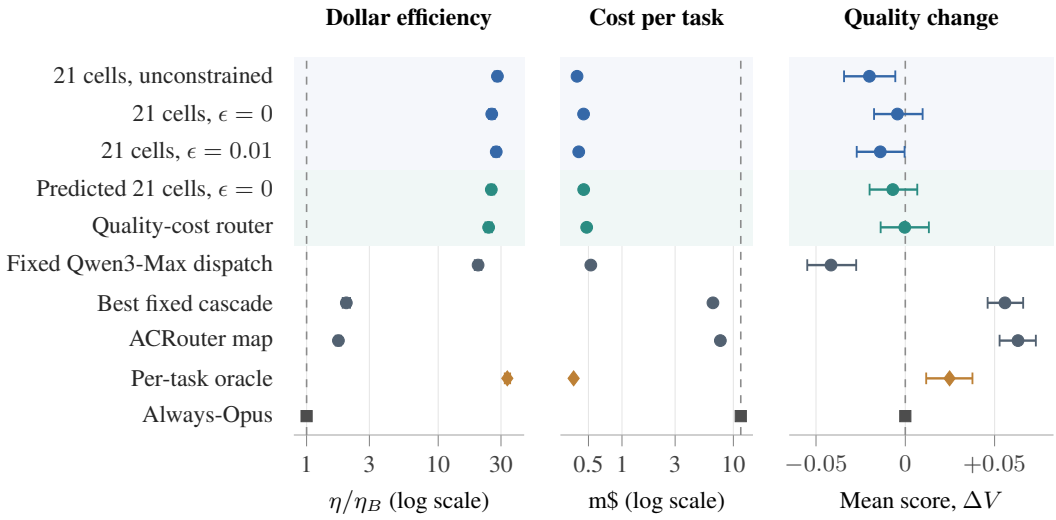


Figure 2: **Efficiency, cost, and quality favor different policies.** Results on the 2,919-task test set with fixed-policy paired 95% intervals. Floors are fitted on probing; the per-task oracle, an in-sample ceiling, gives each test task the action maximizing $V - \eta^\star C$ on its realized outcomes. Dashed lines mark always-Opus. Teal rows use no oracle cell labels: cells predicted from the prompt, and a direct quality-cost router in the style of Dekoninck et al. (2025) at its probing zero-loss point.

Finding: the price structure, not the ratio, removes escalation. The dollar-optimal policies select non-Opus direct dispatch on every task, for both partitions and all fitted floors (Table 7). Proposition 4 explains this: on the 21 probing cells the admission price $\lambda^B = 100$ score per dollar ($2.9\eta_B$) lies below the zero-loss floor price 144 and far below $\eta^\star = 1022$, so every fitted floor asking less than $\delta^B = +0.056$ over Opus is reference-free; without states, every zero-loss policy calls Opus, which runs on 28.7% of tasks. The 21-cell unconstrained policy reaches $28.19\times$ Opus efficiency but loses 0.0201 mean score; the zero-loss floor gives $25.52\times$ and -0.0044. Including policy refitting gives bootstrap medians $25.72\times$ $[23.01, 28.04]$ and $\Delta V = -0.0039$ $[-0.0178, +0.0099]$; the best fixed cascade gains 0.0558 at $2.00\times$. States add 27% over fixed Qwen3-Max dispatch ($20.10\times$ at -0.0416) and recover its 0.042 loss, and a 31% cut in Opus's price would bring it back into the zero-loss policy (Corollary 17). A fitted floor is not a held-out guarantee: refitted on a random half of probing, it keeps test $\Delta V \geq 0$ in only 12.5% of 200 splits. Certifying it on the other half (Proposition 3, empirical-Bernstein p-values) keeps test $\Delta V \geq 0$ in all 200 splits at a median $19.4\times$ and certified price $4.2\eta_B$.

Optimizing mean sequential seconds instead gives $1.22\times$ $[1.18, 1.26]$ time efficiency with $\Delta V = +0.0132$ $[+0.0063, +0.0205]$: the resource changes the selected policy (Appendix A.9).

Robustness: inferred states, learned routers, and the yardstick. Oracle cells are not needed: prompt-only TF-IDF cells (92.9% accuracy) give $25.32\times$ at $\Delta V = -0.0070$ $[-0.0203, +0.0058]$ (teal rows in Figure 2), and after one rule strips source-specific instruction wrappers from every prompt (76.6% accuracy), $24.92\times$ at -0.0032 $[-0.0163, +0.0096]$. Learned quality-cost routers (Dekoninck et al., 2025) and a kNN router reach 22.7–$24.2\times$ at their probing zero-loss points, within 0.006 score of the cell-policy frontier at matched cost, so the floor selects an operating point rather than a better frontier; FrugalGPT- and AutoMix-style cascades hold quality at $11.8\times$ and $1.6\times$. The multiplier depends on the yardstick: sending each cell to its best probing model scores 0.083 above Opus at $2.32\times$; against that yardstick the refitted zero-loss policy gives $3.18\times$ at $\Delta V' = -0.004$ $[-0.011, +0.003]$, whereas the Opus-referenced one falls 0.087 below it (Appendix A.4).

A second benchmark. On RouterBench no reference-free router comes within 0.097 of gpt-4-1106-preview, so the reference is indispensable in the sense of Proposition 4: every zero-loss policy calls it. With the eight source families as states, the $\epsilon = 0$ policy cascades cheap primaries to it and reaches $7.10\times$ at $\Delta V = -0.004$ $[-0.008, +0.001]$. Certified on half of the fitting prompts, the floor holds on held-out prompts in all 200 random halvings at a median $6.70\times$, whereas the uncertified

refit holds in none. Learned quality-cost cascades tie with the 86-dataset state policy at matched cost and beat the 8-family one by 0.014–0.022 (Appendix A.5).

Aggregate versus state-wise guarantees. The floor holds in aggregate for the law that generated the certification sample, not on every subpopulation. The frozen $\epsilon = 0$ cell policy loses 0.054 $[0.015, 0.093]$ on the 334-task verifier sample, and the certified router is negative in a median 8 of 21 test cells; certifying every cell instead (Corollary 22) keeps all cells nonnegative, at $2.34\times$, compared with $1.99\times$ using direct dispatch alone.

3.2 Which Model Adds Cost-Aware Coverage

Decision rule and test. For portfolio S and candidate m, let $S^+ = S \cup \{m\}$. Applying the reference-relative comparison to the current portfolio gives the paired gain

$$G_m(S) = \mathbb{E}[V_{S^+} - V_S] - \eta(S)\mathbb{E}[D_{S^+} - D_S], \qquad \eta(S^+) - \eta(S) = \frac{G_m(S)}{\mathbb{E}[D_{S^+}]}. \tag{7}$$

This identity allows accepted quality and fallback outcomes to change. Under exact selection, let $H_{S,\theta}$ indicate that at least one member passes and define residual coverage as $r_m(S) = \mathbb{P}(H_{m,\theta} = 1, H_{S,\theta} = 0)$. If the compared portfolios share an accepted value v_h, miss-conditional fallback value v_{miss}, and fallback cost b, Proposition 6 reduces the paired gain to

$$N_m(S) = r_m(S)\,[v_h - v_{\text{miss}} + \eta(S)b] - \eta(S)d_m(S), \tag{8}$$

where $d_m(S)$ is the added call, aggregation, and verification cost. Residual coverage is the marginal contribution of a component to the virtual best solver of an algorithm portfolio (Xu et al., 2012; Fréchette et al., 2016); the routing-specific terms are the reference-fallback cost b and the shift of the fallback population. Standalone success does not identify the overlap (Proposition 7); synthesis requires final-answer values in Eq. (7).

Candidate choices are fitted on probing and evaluated on held-out tasks. CodeRouterBench uses seven candidates and portfolios of up to three models; RouterBench adds ten candidates to 21 initial portfolios across contexts. An oracle verifier selects a passing member, else the reference; all member calls are charged. Paired selection picks the candidate with the highest fitted $\hat{\eta}(S^+)$. Controls include marginal success, an independence proxy, and the saving score $r_m b - d_m$ (Appendix A.2).

Table 1: Best-candidate selection experiments on CodeRouterBench and RouterBench. A comparison is one context-portfolio configuration with several candidate additions; an entry is the share of comparisons in which a rule's fitted choice has the highest held-out efficiency among the candidates. Counts sum comparisons across splits, which share tasks. Paired ranks normalized gain in Eq. (7); residual ranks $N_m(S)$; Indep. substitutes independent failures; saving ranks $r_m b - d_m$. RouterBench's $\theta = 1$ row excludes binary contexts.

Dataset, threshold	Splits	Count	Paired	Residual	Indep.	Saving	Marginal
CodeRouterBench, $\theta = 1$	1	139	**97.1%**	86.3%	86.3%	95.7%	46.8%
CodeRouterBench, $\theta = 0.5$	1	194	**96.4%**	94.3%	91.8%	93.3%	40.7%
RouterBench, $\theta = 0.5$	10	5,539	**63.9%**	61.9%	35.9%	63.5%	14.6%
RouterBench, $\theta = 1$	10	212	**64.6%**	53.8%	36.3%	48.1%	0.0%

Finding: frozen conditional means misrank candidates. Residual coverage already prices the avoided fallback cost through $\eta(S)b$, as the saving rule does; the two differ only in the quality term $r_m(S)(v_h - v_{\text{miss}})$, which Eq. (8) evaluates at the current portfolio's conditional means. Those means shift: adding a model lowers the miss-conditional value in 97% of CodeRouterBench comparisons. Adding Qwen3.5-Plus to the pooled Qwen3-Max portfolio, for example, is predicted to lose 1.4 score/\$ but gains $+5.0$ $[3.5, 6.6]$, because fallback cost on the new failure set falls by 14%. The paired rule estimates $\eta(S^+)$ from both portfolios on the same fitting tasks and has the highest hit rate in every setting (Table 1), as a direct estimate should, but by little: at CodeRouterBench $\theta = 1$ it wins 15 of 15 discordant comparisons against residual coverage and 3 of 4 against saving, and every context-cluster bootstrap interval for these differences includes zero, e.g. $[-2.1, +25.6]$ and $[-4.1, +5.4]$ points there. The exact comparison matters most when members are called only after failures: in a cheapest-first cascade at $\theta = 0.5$, paired selection reaches 97.4%, against 88.1% for residual coverage and 86.6% for saving (Appendix A.2).

Regret across splits, sources, and tasks. Across ten CodeRouterBench resplits, paired selection forgoes 0.04–0.06% of the best candidate's efficiency, against 0.18–0.19% for saving and 13–17% for marginal success; on RouterBench paired and saving both forgo 3.6% (Appendix A.2).

4 When Should a Router Spend More?

4.1 What Should Happen After a Failure

Decision rule and test. After a failure, the next action faces a selected subset of tasks. For history $\mathfrak{h}$ and action u, let a be the incremental cost, p_θ the acceptance probability, and v_{acc} the accepted value, all conditional on $(\mathfrak{h}, u)$. Direct fallback has return $J^{\mathrm{direct}}(\mathfrak{h}) = \mathbb{E}[Y_B - \lambda C_B \mid \mathfrak{h}]$. The optimal remaining return satisfies

$$\Pi(\mathfrak{h}) = \max\Big\{ J^{\mathrm{direct}}(\mathfrak{h}),\ \max_u \big[-\lambda a(\mathfrak{h},u) + p_\theta(\mathfrak{h},u) v_{\mathrm{acc}}(\mathfrak{h},u) + (1 - p_\theta(\mathfrak{h},u)) \mathbb{E}[\Pi(\mathfrak{h}') \mid \mathfrak{h}, u, H_\theta = 0] \big] \Big\}. \tag{9}$$

At the horizon, $\Pi = J^{\mathrm{direct}}$. Treating complete continuation plans as actions, Theorem 2 makes Eq. (9) at $\lambda = \eta^\star$ the ratio-optimal continuation, and at $\eta^\star_\epsilon/(1+\kappa)$ its quality-floor version. Proposition 10 gives the shared-value specialization. We fit 259 chains of one to three non-Opus models followed by Opus, at two thresholds and five prices $\lambda \in \{0.1, 1, 10, \eta_B, 100\}$ score per dollar ($\eta_B \approx 34.7$), and compare conditional planning with marginal-success planning. The one-step cost-only rule $p \geq a/b$ is an additional control (Appendix A.6).

Finding: a failure often makes further attempts less attractive. All 42 ordered pairs have lower second-model success after failure at $\theta = 1$ (median ratio 0.49), consistent with shared task difficulty. On 57,405 interval-decided (cell, chain, stage, price) decisions, gain-direction agreement is 93.2% for state-matched planning, 77.2% for marginal planning, and 67.5% for never attempting; the decisions share tasks and chains, so these rates measure how well probing estimates carry over to test. Conditional planning chooses shorter chains in 36 of 105 state-price settings; 27 paired intervals favor it and two favor marginal planning (Appendix A.6).

Online supplement: feedback improves retries. Across seven models on 282 tasks with separable tests, 263 initial failures receive three fresh and three feedback retries. Feedback uses one test half and scoring uses the disjoint half. Mean per-retry recovery is 0.304 with feedback versus 0.165 fresh, a paired gain of 13.9 percentage points $[5.8, 22.3]$ (task-cluster bootstrap), with positive point estimates for all seven models (Appendix A.6). Scoring on the feedback tests themselves overstates repair; this matches cost-aware evaluations of self-repair (Olausson et al., 2024), which compare feedback-driven debugging (Chen et al., 2024d; Shinn et al., 2023) with resampling at equal budget.

4.2 Which Verifier Makes a Cascade Worthwhile?

Decision rule and test. Each verifier-threshold pair is an action with delivered quality and total cost, evaluated by Eq. (4) and the quality floor. The study uses three primaries, 500 probing tasks and 334 runnable test tasks, comparing a static *Compiler* gate, visible examples, hidden tests, a score oracle, and a Grok judge on two primaries. Compiler compiles the extracted answer with the task harness without executing it; hidden tests are an oracle-like control. The judge (grok-4.6) is charged at a primary-price proxy and at its list price ($2/$6 per million input/output tokens), and sandbox executions at their measured CPU time. Figure 3 reports Qwen3-Max (Appendix A.7).

Finding: verification offers a clear cost-quality menu. At $\theta = 0.5$, a free judge yields $2.63\times$ efficiency, remaining at $1.09\times$ under the proxy ($0.30\times$ at the judge's list price) and reaching $0.64\times$ at $\theta = 1$; its Qwen3-Max break-even mean charge is $0.00330 per task, with paired 95% interval [$0.00285,$0.00379]. The static Compiler gate is a zero-dollar protocol check with AUROC 0.766 and $4.09\times$ dollar efficiency at both thresholds ($Q = -0.024$ $[-0.063, +0.009]$ at $\theta = 0.5$). A sandbox check takes 0.15–0.16 s of CPU per task, about $\$5\times10^{-6}$ at a serverless list price and at least 200 times below its break-even charge, so it changes dollar efficiency by at most 0.01. In sequential seconds the ranking reverses, because Opus answers faster than the primaries (3.9 s versus 6.2 s for

Qwen3-Max): verified cascades reach 0.40–$0.54\times$ for Qwen3-Max and Kimi-K2.5, and the judge's 39 s call lowers the ratio to $0.08\times$ (Appendix A.9). Screen with Compiler or tests, which are nearly free in dollars, and choose the resource before the verifier.

Deployable verifiers inside the policy menu. Section 3.1 accepts cascades on the released score for free. Rebuilding that study on this sample with recorded verdicts at measured charges, the pooled zero-loss policy replaces every oracle cascade by a hidden-test cascade and keeps 97% of its efficiency ($3.60\times$ versus $3.72\times$, $\Delta V = -0.003$). Hidden tests are the benchmark's grader; without them only Compiler cascades enter, at $2.33\times$ and $\Delta V = -0.015$ $[-0.045, +0.015]$, whereas direct dispatch meets the pooled floor only by always calling Opus. The judge enters no fitted policy at either price (Appendix A.8).

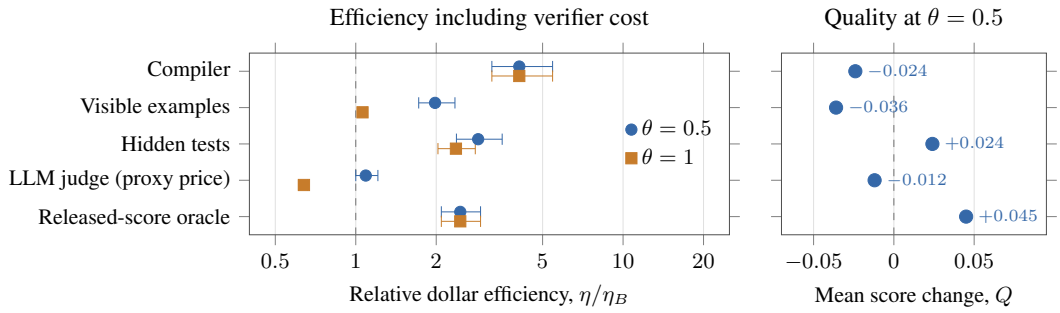


Figure 3: **Qwen3-Max verifier tradeoffs.** Relative efficiency (left) and quality change at $\theta = 0.5$ (right), with paired 95% intervals. Judge fees use proxy pricing.

4.3 Is Routing Information Worth Buying?

Decision rule. In the quality-preserving two-action model, attempting costs $C_{\text{attempt}}(s) = a + (1 - s)b$, where s is the primary's success probability, a its cost, and b the fallback cost, with $0 < a < b$; direct fallback costs b. If a signal costing k changes the posterior to s_Z, its net value is

$$\min\{a + (1 - s)b, b\} - \mathbb{E}\min\{a + (1 - s_Z)b, b\} - k. \tag{10}$$

Proposition 9 bounds gross savings by perfect information. We report *capture*, gross savings divided by that bound, before subtracting the signal charge.

Test 1: what information do released routers supply? Nine released routers expose model choices that we use as binary signals, with posterior success estimated by ten half-split cross-fits. Pooled tasks supply no type label; conditioning supplies the benchmark dimension as oracle task-type metadata. Signals are evaluated at zero signal price (Appendix A.10).

Finding: released routers mainly recover task type. Under posterior-threshold use, the two largest of four fine-tuned routers capture **54.6–55.5%** of the pooled bound, versus **6.2–7.8%** of the dimension-conditioned bound (Table 15 in Appendix A.10). Since real tasks rarely carry type labels, inferring task structure is itself a major source of routing value.

Test 2: what does a signal cost? Proposition 9 gives a condition that needs no accuracy estimate: information pays only if its price k is below the attempt cost a. Measured prices separate signals sharply. A prompt-only TF-IDF router takes about 2 ms of CPU per task, 0.06% of a Qwen3-Max attempt, and predicts Qwen3-Max and Kimi-K2.5 success with held-out AUROC 0.895 and 0.768. A DeepSeek-v4-pro forecast (1,200 live calls) averages 635 tokens and 1.9 s and at list price costs 2.4 Qwen3-Max attempts, exceeding a in 22 of 24 states whatever its accuracy (Appendices A.10 and A.9).

5 What Should a Router Remember?

Question and decision rule. Retained choices must improve on the current prior. Let cell j arrive with probability ν_j, have stationary coverage χ_j, and erroneous write fraction e_j. For an informative or erroneous state $s \in \{I, W\}$, define $g_j^s = (v_j^s - v_j^U) - \eta_U(c_j^s - c_j^U)$ relative to prior U with the same exploration and writing. Here v_j^s, c_j^s are the cell- and state-conditional value and cost, and η_U is the prior's efficiency. Under the fixed write process, Proposition 11 gives

$$\mathbb{E}[V - \eta_U D] = \sum_j \nu_j \chi_j \left[(1 - e_j) g_j^I + e_j g_j^W\right] \tag{11}$$

at stationarity. Positive surplus means an efficiency gain. If $g_j^I > 0 > g_j^W$, a covered cell helps when $e_j < g_j^I/(g_j^I - g_j^W)$ (the quality–cost comparison in Figure 1c).

Test: observed outcomes, constructed record streams. This **semi-synthetic** study combines real outputs and verifier judgments with constructed arrivals and retention: 2,044 probing and 867 test tasks in 12 cells. Exploration occurs with probability 0.25; acceptance writes a model identity, which serves its cell, and a write is erroneous when the accepted output is incorrect under the released score. Fifty streams of 60,000 rounds are measured at stationarity (Appendix A.11).

Finding: the prior determines whether memory helps. With hidden-test writes and no expiration, the store reaches 135.8 score/$ but scores 0.034 below always-Opus and 0.116 below DimensionBest, so as a router it does not meet a zero-loss floor. Against always-Opus, the probing-predicted per-round gain is $+0.304$ and the realized gain $+0.275$ $[0.272, 0.279]$; against a probing-fitted DimensionBest prior, Eq. (11) predicts the opposite sign, and the realized gain has it (Appendix A.11). Cell-level sign agreement is 918/969 (94.7%), versus 646/969 (66.7%) for always predicting improvement, and injected writer errors leave the aggregate sign unchanged. Memory value depends on the prior and writer, not coverage alone; predicted gain magnitudes are less accurate than their signs (Appendix A.11).

6 Discussion

Takeaways. A routing decision is worthwhile when its added value justifies its full cost within the quality requirement. In practice: **1)** fix the quality requirement and resource before ranking policies, and compare complete policies on matched tasks; **2)** evaluate continuation on the tasks that reach it; **3)** value a verifier, signal, or record against the prior it replaces, at its real price.

Relation to existing work. Dekoninck et al. (2025) characterize optimal routing and cascading through estimated quality-cost tradeoffs, and Jitkrittum et al. (2023) show why deferral must account for downstream performance. Learned routers (Ding et al., 2024; Chen et al., 2024c; Feng et al., 2025; Zhuang et al., 2025; Ding et al., 2025) estimate the per-model quality and cost that our menus take as given; the tools we apply, fractional programming, parametric linear programming, and fixed-sequence testing, are classical. What the accounting adds is the choice of the operating point by a reference and tolerance, a finite-sample certificate for it, and matched comparisons that expose fallback-population shifts (Appendix N).

Limitations. The theory does not cover policy-dependent state transitions: Theorem 2 and Proposition 3 take the state law as given, and the memory results evaluate a given writer. Floor certification is offline and needs the reference's outcome on every certification task, so a deployment pays to run the reference there and efficiency ratios also depend on the price table and the reference.

7 Conclusion

Reference-relative accounting connects what a router combines, when it spends more, and what it remembers. In model routing it shows that fallback populations, information charges, and the existing prior decide which choice pays, and it separates efficiency gains from quality preservation. Together, these results provide a practical basis for choosing routing policies under explicit quality requirements and evaluating added computation on matched tasks at measured cost.

AI USE STATEMENT

Large language models assisted with manuscript drafting and reviewing. The authors drafted and verified all the content and take full responsibility for the paper, including its text, claims, experiments and attached artifacts.

ETHICS STATEMENT

Lower inference cost may improve access to capable systems, while extra calls and verification consume resources. We report quality separately from efficiency and identify cases where routing or erroneous records reduce quality. Downstream applications must evaluate their own quality requirements and measured costs.

REPRODUCIBILITY STATEMENT

The appendices contain proofs, analytical illustrations, and experimental protocols and results. Supporting materials record data provenance and analysis procedures.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 131000–131034, 2024.

Eitan Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1999.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998.

Fu Bang. GPTCache: An open-source semantic cache for LLM applications enabling faster answers and cost savings. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pp. 212–218, 2023.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):43:1–43:34, 2021. doi: 10.1145/3478535.

Frederick J Beutler and Keith W Ross. Optimal policies for controlled markov chains with a constraint. *Journal of mathematical analysis and applications*, 112(1):236–252, 1985.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24 (2):265–272, 1953. doi: 10.1214/aoms/1177729032.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Abraham Charnes and William W Cooper. Programming with linear fractional functionals. *Naval Research logistics quarterly*, 9(3-4):181–186, 1962.

Hao Che, Ye Tung, and Zhijun Wang. Hierarchical Web caching systems: Modeling, design and experimental results. *IEEE Journal on Selected Areas in Communications*, 20(7):1305–1314, 2002. doi: 10.1109/JSAC.2002.801752.

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.

Lingjiao Chen, Jared Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. Are more LLM calls all you need? towards the scaling properties of compound AI systems. In *Advances in Neural Information Processing Systems*, volume 37, pp. 45767–45790, 2024a.

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*, 2024b. ISSN 2835-8856.

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*, volume 37, pp. 66305–66328. Curran Associates, Inc., 2024c. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7a641b8ec86162fc875fb9f6456a542f-Abstract-Conference.html.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*, 2024d. URL https://openreview.net/forum?id=KuPixIqPiq.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130185–130213, 2024e.

Chao Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970.

Jeffrey Dean and Luiz André Barroso. The tail at scale. *Communications of the ACM*, 56(2):74–80, 2013.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 12987–13010, 2025.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. BEST-route: Adaptive LLM routing with test-time optimal compute. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 13870–13884. PMLR, 2025. URL https://proceedings.mlr.press/v267/ding25d.html.

Werner Dinkelbach. On nonlinear fractional programming. *Management science*, 13(7):492–498, 1967.

Ronald Fagin. Asymptotic miss ratios over independent references. *Journal of Computer and System Sciences*, 14(2):222–250, 1977.

Tao Feng, Yanzhen Shen, and Jiaxuan You. GraphRouter: A graph-based router for LLM selections. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eU39PDsZtT.

Alexandre Fréchette, Lars Kotthoff, Tomasz Michalak, Talal Rahwan, Holger Hoos, and Kevin Leyton-Brown. Using the Shapley value to analyze algorithm portfolios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Tianyu Fu, Yi Ge, Yichen You, Enshu Liu, Zhihang Yuan, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. R2R: Efficiently navigating divergent reasoning paths with small-large model token routing. In *Advances in Neural Information Processing Systems*, volume 38, pp. 124108–124145, 2026a.

Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-cache: Direct semantic communication between large language models. In *International Conference on Learning Representations*, volume 2026, pp. 43130–43158, 2026b.

Luis Gaspar Schroeder, Aditya Desai, Alejandro Cuadron Lafuente, Kyle Chu, Shu Liu, Mark Zhao, Stephan Krusche, Alfons Kemper, Matei Zaharia, and Joseph E Gonzalez. vCache: Verified semantic prompt caching. In *International Conference on Learning Representations*, volume 2026, pp. 92634–92661, 2026.

Saul Gass and Thomas Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2(1–2):39–45, 1955. doi: 10.1002/nav.3800020106.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Alexander V. Gnedin, Ben B. Hansen, and Jim Pitman. Notes on the occupancy problem with infinitely many boxes: general asymptotics and power laws. *Probability Surveys*, 4:146–171, 2007.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Carla P Gomes and Bart Selman. Algorithm portfolios. *Artificial Intelligence*, 126(1-2):43–62, 2001.

Irving J Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264, 1953.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *International Conference on Learning Representations*, volume 2024, pp. 4147–4180, 2024.

Nicholas Hay, Stuart Russell, David Tolpin, and Solomon Eyal Shimony. Selecting computations: Theory and applications. *arXiv preprint arXiv:1408.2048*, 2014.

Taekyung Heo, Rasoul Shafipour, Ritchie Zhao, Maximilian Golub, Mohammad Mahdi Kamani, Ritika Borkar, Makesh Tarun Chandran, Pantea Zardoshti, and Bita Darvish Rouhani. Cross-model KV cache transfer in LLM families: A closed-form linear mapping for prefill reuse. *arXiv preprint arXiv:2608.03893*, 2026.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Ronald A Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966.

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.

Bernardo A Huberman, Rajan M Lukose, and Tad Hogg. An economics approach to hard computational problems. *Science*, 275(5296):51–54, 1997.

Predrag R. Jelenković. Asymptotic approximation of the move-to-front search cost distribution and least-recently used caching fault probabilities. *The Annals of Applied Probability*, 9(2):430–464, 1999. doi: 10.1214/aoap/1029962750.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, 2023.

Wittawat Jitkrittum, Neha Gupta, Aditya K Menon, Harikrishna Narasimhan, Ankit Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice? In *Advances in Neural Information Processing Systems*, volume 36, pp. 9891–9906, 2023.

Jaehun Jung, Faeze Brahman, and Yejin Choi. Trust or escalate: LLM judges with provable guarantees for human agreement. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=UHPnqSTBPO.

Samuel Karlin. Central limit theorems for certain infinite urn schemes. *Journal of Mathematics and Mechanics*, 17(4):373–401, 1967.

Lars Kotthoff. Algorithm selection for combinatorial search problems: A survey. In *Data mining and constraint programming: Foundations of a cross-disciplinary approach*, pp. 149–190. Springer, 2016.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

Haiji Liang, Pengfei Zhou, Zhenglin Wan, Wei Wang, Yang You, and Wangbo Zhao. Beyond one-size-fits-all: Sample-adaptive strategy routing for vision token pruning in MLLMs. *arXiv preprint arXiv:2609.10346*, 2026.

Xinchen Liu, Hang Zhou, Yingjie Zong, Yuchuan Tian, Liuyang Song, Shuo Zhang, Yulong Li, Wei He, Mengyu Zheng, Runke Liu, et al. Agentic routing: The harness-native data flywheel. *arXiv preprint arXiv:2607.11399*, 2026.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009. URL https://arxiv.org/abs/0907.3740.

Willi Maurer, Ludwig A. Hothorn, and Walter Lehmacher. Multiple comparisons in drug clinical trials and preclinical assays: a-priori ordered hypotheses. In Joachim Vollmar (ed.), *Biometrie in der chemisch-pharmazeutischen Industrie*, volume 6, pp. 3–18. Gustav Fischer Verlag, Stuttgart, 1995.

Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Seungyeon Kim, Neha Gupta, Aditya Krishna Menon, and Sanjiv Kumar. Faster cascades via speculative decoding. In *International Conference on Learning Representations*, volume 2025, pp. 44949–44987, 2025.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Stefan Nielsen, Edoardo Cetin, Peter Schwendeman, Qi Sun, Jinglue Xu, and Yujin Tang. Learning to orchestrate agents in natural language with the conductor. In *International Conference on Learning Representations*, volume 2026, pp. 135686–135724, 2026.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=y0GJXRungR.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Xingyu Qu, Siyuan Lu, Zhiyu Chen, Sheng Wang, and Tao Lin. CacheBridge: Efficient cross-model KV cache transfer. *arXiv preprint arXiv:2609.00891*, 2026.

Howard Raiffa and Robert Schlaifer. *Applied Statistical Decision Theory*. Division of Research, Graduate School of Business Administration, Harvard University, 1961.

John R Rice. The algorithm selection problem. In *Advances in computers*, volume 15, pp. 65–118. Academic Press, 1976.

Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial intelligence*, 49(1-3):361–395, 1991.

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.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025.

Benedikt Stroebl, Sayash Kapoor, and Arvind Narayanan. The limits of inference scaling through resampling. In *International Conference on Learning Representations*, volume 2026, pp. 92601–92633, 2026.

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

Yujin Tang, Edoardo Cetin, Jinglue Xu, Qi Sun, Stefan Nielsen, Vincent Richard, Haruto Goda, Iaroslav Tymchenko, Nhan Nguyen, Hyunin Lee, et al. Sakana fugu technical report. *arXiv preprint arXiv:2606.21228*, 2026.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2380–2388. PMLR, 2015. URL https://proceedings.mlr.press/v37/thomas15.html.

Philip S. Thomas, Bruno Castro da Silva, Andrew G. Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019. doi: 10.1126/science.aag3311.

Kirill Trapeznikov and Venkatesh Saligrama. Supervised sequential classification under budget constraints. In *Artificial intelligence and statistics*, pp. 581–589, 2013.

Rajeev Verma, Daniel Barrejón, 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*, pp. 11415–11434, 2023.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025.

Peter H. Westfall and Alok Krishen. Optimally weighted, fixed sequence and gatekeeper multiple testing procedures. *Journal of Statistical Planning and Inference*, 99(1):25–40, 2001. doi: 10.1016/S0378-3758(01)00077-5.

Jinglue Xu, Qi Sun, Peter Schwendeman, Stefan Nielsen, Edoardo Cetin, and Yujin Tang. Trinity: An evolved LLM coordinator. In *International Conference on Learning Representations*, volume 2026, pp. 70195–70224, 2026.

Lin Xu, Holger Hoos, and Kevin Leyton-Brown. Hydra: Automatically configuring algorithms for portfolio-based selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 210–216, 2010.

Lin Xu, Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. Evaluating component solver contributions to portfolio-based algorithm selectors. In *International conference on theory and applications of satisfiability testing*, pp. 228–241, 2012.

Tao Yu, Yifei Qu, Zhiqing Cui, Pengfei Zhou, Zhongtian Luo, Yujia Yang, Shenghua Chai, Haopeng Jin, Zhenghao Zhang, Xinming Wang, et al. SCOPE-Router: Cost-aware open-set VLM routing for execution-oriented tasks. *arXiv preprint arXiv:2608.12127*, 2026.

Pengfei Zhou, Zhiwei Tang, Yixing Ma, Jiasheng Tang, Yizeng Han, Zhenglin Wan, Fanqing Meng, Wei Wang, Bohan Zhuang, Wangbo Zhao, et al. Agent-as-a-Router: Agentic model routing for coding tasks. *arXiv preprint arXiv:2606.22902*, 2026.

Banghua Zhu, Ying Sheng, Lianmin Zheng, Clark Barrett, Michael Jordan, and Jiantao Jiao. Towards optimal caching and model selection for large model inference. In *Advances in Neural Information Processing Systems*, volume 36, pp. 59062–59094, 2023.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Fs9EabmQrJ.

Jiaru Zou, Ruizhong Qiu, Gaotang Li, Xiyuan Yang, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, et al. Latent collaboration in multi-agent systems. *arXiv preprint arXiv:2511.20639*, 2025.

APPENDIX ROADMAP

Appendix A gives experimental details, and Appendices B-M the formal results and proofs. Table 2 locates each numbered result.

Table 2: Where each numbered result is stated and proved.

Result	Statement	Proof
Lemma 1	Sec. 2	App. G.1
Theorem 2	Sec. 2	App. H.3
Proposition 3	Sec. 2	App. K.1
Proposition 4	Sec. 2	App. H.4
Proposition 21, Corollary 22	App. K.2	in place
Propositions 5-7	App. C	App. L
Proposition 24	App. L.4	App. L.5
Propositions 8, 9, 10	App. D	App. H
Corollary 19	App. H.6	in place
Proposition 14	App. G.7	in place
Propositions 11, 12	App. E	App. M.1, M.2
Proposition 13, Corollary 27	App. E.1	App. M.7, M.8
Proposition 20	App. J.1	App. J.2

A EXPERIMENTAL DETAILS

A.1 DATA, ACCOUNTING, AND ESTIMATION

We use the released CodeRouterBench outcome matrix and price list (Zhou et al., 2026): 9,999 coding tasks, eight models, nine dimensions, and 21 non-empty dimension×source cells. The release splits these tasks into probing (7,080; train and validation combined) and a test split (2,919). The models are claude-opus-4-6, claude-sonnet-4-6, gpt-5.4, Qwen3-Max, Kimi-K2.5, Qwen3.5-Plus, glm-5, and MiniMax-M2.7. The reference, always-Opus, has mean test score 0.4383 and total estimated cost $34.02. RouterBench uses the released 0-shot matrix of 36,497 prompts and 11 models (Hu et al., 2024).

Replay and online measurements. Unless stated otherwise, a replay accepts the primary when its released score is at least $\theta \in \{1, 0.5\}$, then uses the recorded Opus outcome on a miss: $V = HY_P + (1-H)Y_B$ and $D = C_P + (1-H)C_B$. This is an oracle-verifier replay with no verifier charge. Dollar costs are token-price estimates; router overhead is excluded unless explicitly priced. Sequential seconds add recorded primary and fallback latencies. The online retry, verifier, paid-forecast, and memory studies include fresh calls through one OpenAI-compatible API relay, but all Opus and Sonnet outcomes remain recorded; those comparisons combine online calls with replayed reference outcomes. Requests name the model (the grok-4.6 judge without extended reasoning and with 1,024 completion tokens; the `deepseek-v4-pro` forecast at temperature 0); the relay reports no served version.

Evidence types and charged costs. Table 3 lists, for every study, what was executed live, what was replayed, how acceptance was decided, and which costs enter D. *Replay* rows evaluate the decision rule on recorded outcomes with oracle acceptance; *online* rows add fresh calls; in every study the Opus fallback is a recorded outcome. Appendix A.9 measures the router, verifier, and latency charges that replay excludes.

Samples and intervals. Primitives and policy choices are fitted on probing and evaluated on the test set, except the explicitly cross-fitted signal analysis below. Intervals are 95% percentile task-bootstrap intervals, paired for policy comparisons, with 500 or 1,000 resamples unless specified; record simulations also resample streams. A sign is decided when its realized interval excludes zero. Configuration counts reuse tasks. Efficiency is the ratio of aggregate value to aggregate resource, not the average of task-wise ratios.

Table 3: Evidence type, executed components, acceptance, and charged costs of each study. Model calls are charged at recorded token-price costs throughout.

Study (section)	Type	Live calls	Acceptance	Router, verifier, latency charges
Policy selection (3.1, A.3–A.4)	replay	none	released score for cascades; none for dispatch	router uncharged (measured below 0.1% of a call); recorded latencies in seconds
Portfolio selection (3.2, A.2)	replay	none	oracle selecting verifier	every member call; verifier uncharged
Failure-conditioned chains (4.1, A.6)	replay	none	released score	verifier uncharged
Online retries (4.1, A.6)	online	1,415 chains; expanded seven-model retries	hidden tests (disjoint feedback and scoring halves for retries)	retry tokens at list or proxy prices; sandbox time in seconds
Verifier operating points (4.2, A.7)	online	1,668 judge calls	Compiler, visible tests, hidden tests, judge, released score	judge tokens at proxy and list prices; measured sandbox and judge time
Released signals (4.3, A.10)	replay	none	two-action cost model	signal at zero price
Paid forecast (4.3, A.10)	online	1,200 forecasts	threshold a/b	forecast tokens at list price; call latency
Memory (5, A.11)	semi-synthetic	judge on a subsample	doctests, hidden tests, or judge	verification uncharged in dollars

A.2 Residual coverage on two datasets

An oracle selecting verifier commits to the best passing portfolio member and uses the reference on a miss; costs sum member calls, with no aggregation fee. CodeRouterBench uses the seven non-Opus candidates, portfolios of up to three members, and nine-dimensional contexts. A candidate needs at least 20 passes and 20 failures, and a non-trivial comparison at least ten tasks in each conditional cell. RouterBench uses gpt-4-1106-preview as reference, ten candidate additions to 21 one- or two-member portfolios, and a grid of eight source families and 57 MMLU subsets. The main candidate-selection results pool ten fixed 70/30 splits, each with 25,542 fitting and 10,955 held-out prompts, stratified by source and reference success. Filtering and ranking use fitting tasks only. Across these splits, filtering retains 5,539 comparisons at $\theta = 0.5$ and 212 in nonbinary contexts at $\theta = 1$. Sign and outcome diagnostics below use the original split.

The score $N_m(S) = r_m(S)[v_h - v_{\text{miss}} + \eta(S)b] - \eta(S)d_m$ uses the candidate's residual coverage and incremental cost. The saving baseline ranks $r_m(S)b - d_m$, with b estimated on failures of the original portfolio; it values avoided fallback work without an explicit quality term. The independence baseline replaces $r_m(S)$ with $(1 - p(S))\mathbb{P}(E_m)$; other controls rank marginal success or residual coverage alone. On the original RouterBench split at $\theta = 0.5$, we correlate each fitting score with held-out efficiency gain across candidates within a context-portfolio comparison, then take the median across 566 comparisons: 0.928 for $N_m(S)$, 0.080 for residual coverage alone, and -0.412 for marginal success. These correlations measure full candidate rankings, whereas Table 5 evaluates the selected candidate's delivered outcomes.

The common-conditional-value assumption is imperfect on both datasets. Adding a CodeRouterBench model lowers miss-conditional value in 97% of the comparisons, with median changes of -13.7% and -17.5% at the two thresholds. In the pooled Qwen3-Max portfolio, adding Qwen3.5-Plus is predicted to lose 1.4 score per dollar but gains 5.0 $[3.5, 6.6]$, because the new failure set lowers fallback cost by 14%. On its original split, RouterBench shows conditional-value violations in 56% of 6,392 comparisons and median absolute relative magnitude error of 7.4%. The paired comparison below removes this approximation.

Paired reanalysis without fixed conditional means. The following sign and outcome diagnostics use CodeRouterBench's official split and RouterBench's original split. Parameters and selected candidates use fitting tasks only. For sign prediction we use $\widehat{G}_m(S)$; for candidate selection we divide

by $\widehat{\mathbb{E}}[D_{S^+}]$, because positive gains alone do not rank efficiency changes. The held-out best candidate evaluates the fixed fitting-split choice.

Table 4: Gain-direction agreement on all held-out comparisons, using the official CodeRouterBench and original RouterBench splits. Saving predicts the sign of $r_m b - d_m$; always-negative predicts that every addition lowers efficiency. The $\theta = 1$ RouterBench control includes only nonbinary contexts.

Dataset and threshold	Comparisons	Paired	Residual	Indep.	Saving	Always-negative
CodeRouterBench, $\theta = 1$	674	**93.8%**	91.4%	71.5%	81.6%	61.9%
CodeRouterBench, $\theta = 0.5$	958	**94.8%**	93.1%	71.2%	85.4%	66.9%
RouterBench, $\theta = 0.5$	3,602	**93.4%**	93.2%	85.9%	92.9%	44.6%
RouterBench, $\theta = 1$	176	**97.7%**	96.6%	80.7%	94.3%	42.6%

We draw 500 paired task-bootstrap resamples per context, holding the fitting scores and selected candidates fixed. Among interval-decided additions, paired versus residual sign agreement is 100.0% versus 96.6% on 522 CodeRouterBench comparisons at $\theta = 1$, 99.6% versus 98.6% on 775 at $\theta = 0.5$, and 98.5% versus 98.2% on 2,548 RouterBench comparisons at $\theta = 0.5$. Table 4 reports all comparisons, without selecting by interval. In the nonbinary RouterBench control, paired versus residual sign agreement is 97.7% versus 96.6% over 176 additions.

The paired rule changes 15 CodeRouterBench selections at $\theta = 1$; all 15 have higher held-out efficiency, and eight paired difference intervals exclude zero. At $\theta = 0.5$, six selections change: five improve and one worsens, with one improvement interval above zero.

Uncertainty of best-candidate frequencies. With fitted choices held fixed, we resample held-out outcomes and recompute the held-out best candidate in each resample. Because comparisons share tasks and base portfolios, we resample clusters of comparisons (CodeRouterBench dimensions; RouterBench source families, with MMLU subsets inside MMLU) together with their held-out tasks, 2,000 times. Task-only resampling is not reported: re-identifying the best candidate on resampled outcomes can only lower a rule's hit rate, so those percentile intervals end at the point estimate. Differences in best-candidate frequency (percentage points), cluster intervals, and discordant comparisons are:

Setting	Paired − saving	Paired − residual
CodeRouterBench, $\theta = 1$ (139)	$+1.4$ $[-4.1, +5.4]$; 3 vs. 1	$+10.8$ $[-2.1, +25.6]$; 15 vs. 0
CodeRouterBench, $\theta = 0.5$ (194)	$+3.1$ $[-1.0, +9.3]$; 8 vs. 2	$+2.1$ $[-2.3, +5.3]$; 5 vs. 1
RouterBench, $\theta = 0.5$ (5,539)	$+0.4$ $[-0.9, +3.9]$; 215 vs. 194	$+2.0$ $[-0.5, +7.2]$; 321 vs. 212
RouterBench, $\theta = 1$ (212)	$+16.5$ $[-7.5, +27.4]$; 56 vs. 21	$+10.8$ $[-15.1, +25.0]$; 53 vs. 30

Every interval includes zero. The RouterBench $\theta = 1$ comparisons come from two source families, so that interval is crude, and discordant counts pooled over RouterBench splits count shared tasks repeatedly. Across RouterBench splits, frequencies are 63.9 ± 1.9 (paired), 63.5 ± 2.7 (saving), and 61.9 ± 2.3 (residual). At CodeRouterBench $\theta = 1$, paired minus saving changes delivered score by -0.00029 $[-0.00053, -0.00004]$, cost by -0.0066 m\$ $[-0.017, +0.003]$, and regret by -0.08 points $[-0.23, +0.08]$; against residual it lowers cost by 0.061 m\$ $[0.024, 0.099]$ and regret by 0.56 points $[0.11, 0.97]$ (task bootstrap, fitted choices fixed). All 15 paired-only wins over residual lie in the upper two thirds of the Eq. (25) correction size, but saving recovers 13 of them without the correction. A non-oracle-charging variant calls members cheapest first and pays only for called members; there paired reaches 97.4% versus 88.1% for residual and 86.6% for saving at CodeRouterBench $\theta = 0.5$ (18 vs. 0 and 22 vs. 1 discordant) and 55.4% versus 48.9% for saving on RouterBench $\theta = 0.5$, partly because the saving and residual scores still charge every candidate's full cost.

Decision outcomes and a cost-aware control. Table 5 retains the original-split outcome diagnostics. We report delivered score, cost, and relative efficiency regret, $(\eta_{\text{best}} - \eta_{\text{selected}})/\eta_{\text{best}}$, where the best candidate is identified on held-out outcomes. Averages weight each context-portfolio configuration equally. On CodeRouterBench at $\theta = 1$, paired selection saves 0.0066 m\$ per task relative to the saving rule while reducing mean score by 0.00029; at $\theta = 0.5$, it adds 0.00048 score for 0.00084 m\$.

Table 5: Outcomes of probing-selected portfolio additions on the official CodeRouterBench and original RouterBench splits. Cost is m$ per task; regret is a percentage of the best held-out candidate's efficiency. RouterBench's $\theta = 1$ row uses nonbinary contexts only.

Dataset	θ	Rule	Mean score	Cost	Regret (%)
CodeRouterBench	1	Paired	0.6110	15.2882	0.060
		Residual	0.6116	15.3488	0.624
		Saving	0.6113	15.2948	0.138
		Marginal success	0.6173	17.8735	12.743
	0.5	Paired	0.5969	12.7720	0.025
		Residual	0.5967	12.7741	0.066
		Saving	0.5964	12.7711	0.103
		Marginal success	0.6052	15.4071	16.191
RouterBench	0.5	Paired	0.8832	0.6945	3.347
		Residual	0.8826	0.6916	3.469
		Saving	0.8823	0.6920	3.396
		Marginal success	0.8739	1.5742	43.508
	1	Paired	0.6656	2.6656	0.946
		Residual	0.6698	2.7208	2.245
		Saving	0.6676	2.7512	3.632
		Marginal success	0.6795	4.4148	38.962

Regret across splits, sources, and tasks. For a comparison with fitted choice $\hat{m}$, *relative regret* is $[\max_m \eta(S_m^+) - \eta(S_{\hat{m}}^+)]/\max_m \eta(S_m^+)$ on held-out tasks, and *task-level regret* is $\max_m G_m(S) - G_{\hat{m}}(S)$, the held-out net score per task lost against the best candidate at the incumbent's held-out price $\eta(S)$ in Eq. (7); task-level averages weight each comparison by its number of held-out tasks. CodeRouterBench adds ten random resplits that keep each dimension's released fitting count (seeds 20260924–20260933); RouterBench uses the ten splits of Table 1. Filters, candidates, and rules are unchanged, and the official split reproduces Table 5.

Table 6: Selection regret across ten splits (mean $\pm$ standard deviation over splits). Task-level regret is in 10^{-3} net score per held-out task.

	CodeRouterBench resplits		RouterBench splits	
Rule	$\theta = 1$	$\theta = 0.5$	$\theta = 0.5$	$\theta = 1$
Relative regret (%)				
Paired	$\mathbf{0.04 \pm 0.03}$	0.06 ± 0.02	3.56 ± 0.34	$\mathbf{0.94 \pm 0.72}$
Saving	0.19 ± 0.08	0.18 ± 0.06	$\mathbf{3.54 \pm 0.40}$	1.68 ± 1.11
Residual	0.29 ± 0.12	$\mathbf{0.05 \pm 0.05}$	3.71 ± 0.36	1.36 ± 1.03
Independence	0.20 ± 0.11	0.15 ± 0.09	8.45 ± 0.63	2.06 ± 0.93
Marginal	13.08 ± 0.44	16.98 ± 0.64	40.04 ± 2.00	35.98 ± 1.89
Task-level regret (10^{-3} net score per task)				
Paired	$\mathbf{0.32 \pm 0.25}$	0.51 ± 0.18	$\mathbf{5.90 \pm 1.12}$	$\mathbf{5.72 \pm 4.47}$
Saving	1.45 ± 0.58	1.18 ± 0.39	7.54 ± 2.12	10.33 ± 7.08
Residual	2.61 ± 1.10	$\mathbf{0.41 \pm 0.39}$	8.67 ± 1.35	8.27 ± 6.54
Independence	1.53 ± 0.70	1.10 ± 0.54	38.2 ± 3.8	12.6 ± 5.9
Marginal	136.4 ± 5.1	172.4 ± 9.5	927 ± 130	331 ± 34

Across splits, paired selection has lower relative and task-level regret than saving and residual coverage in all ten CodeRouterBench resplits at $\theta = 1$; at $\theta = 0.5$, where residual coverage is also near-optimal, the two differ by 0.01 points on average. On RouterBench at $\theta = 0.5$, paired and saving tie in relative regret (paired is lower in 4 of 10 splits) while paired has lower task-level regret in 9 of 10; at $\theta = 1$ it is lower than saving in 8 of 10. Across sources, on the official CodeRouterBench split paired's regret is the lowest or tied in 6 of 7 dimensions at $\theta = 1$ (the exception, multi-language, has three comparisons) and in all 8 at $\theta = 0.5$. Pooling RouterBench's ten splits at $\theta = 0.5$, paired has the lowest task-level regret in 5 of 7 source families; MMLU and MBPP carry the largest regret for paired, saving, and residual alike (3.8–4.6%). Marginal success forgoes 13–40% of the best candidate's efficiency in every setting.

A.3 COMPLETE-POLICY OUTCOMES

Table 7 gives the numerical values underlying Figure 2. The policy menu, fitting procedure, and held-out evaluation are described in Section 3.1.

Table 7: Dollar efficiency, cost, and quality on the test set. $\Delta V = \mathbb{E}[V - Y_B]$; costs are m$ per task. Brackets are paired 95% intervals. Quality floors are fitted on probing; $\epsilon \in \{0.02, 0.05\}$ leave the fitted floor slack and coincide with the unconstrained policy of the same partition. Fixed dispatch is the best single model on probing, which is also the cheapest (Qwen3-Max).

Policy	η/η_B [CI]	Cost	ΔV [CI]
21 cells, unconstrained	28.187 [26.840, 29.659]	0.395	-0.0201 [-0.0343, -0.0056]
21 cells, $\epsilon = 0$	25.516 [24.278, 26.903]	0.452	-0.0044 [-0.0175, $+0.0097$]
21 cells, $\epsilon = 0.005$	27.071 [25.786, 28.538]	0.422	-0.0083 [-0.0212, $+0.0056$]
21 cells, $\epsilon = 0.01$	27.612 [26.311, 29.072]	0.409	-0.0140 [-0.0272, -0.0004]
9 dimensions, unconstrained	28.024 [26.713, 29.452]	0.398	-0.0192 [-0.0332, -0.0047]
9 dimensions, $\epsilon = 0$	22.733 [21.584, 23.954]	0.502	-0.0090 [-0.0227, $+0.0060$]
9 dimensions, $\epsilon = 0.005$	24.241 [23.042, 25.528]	0.468	-0.0116 [-0.0255, $+0.0033$]
9 dimensions, $\epsilon = 0.01$	25.984 [24.735, 27.349]	0.434	-0.0141 [-0.0279, $+0.0004$]
Fixed Qwen3-Max dispatch	20.097 [18.965, 21.297]	0.525	-0.0416 [-0.0549, -0.0275]
Best fixed cascade	2.004 [1.881, 2.140]	6.556	$+0.0558$ [$+0.0461$, $+0.0660$]
ACRouter released map	1.745 [1.666, 1.826]	7.643	$+0.0631$ [$+0.0528$, $+0.0731$]
Per-task oracle	33.510 [32.065, 35.206]	0.368	$+0.0248$ [$+0.0117$, $+0.0376$]
Always-Opus	1.000	11.655	0

Uncertainty with policy refitting. Table 7 holds each fitted policy fixed. To include policy-selection uncertainty, each of 500 replicates randomly halves the 7,080 probing tasks, bootstraps both halves with replacement, refits the quality-floor policy on their union, and evaluates it on an independent bootstrap resample of the 2,919 test tasks. For 21 cells and $\epsilon = 0$, the bootstrap median efficiency is $25.72\times$ with 95% percentile interval $[23.01, 28.04]$; the median ΔV is -0.0039 $[-0.0178, +0.0099]$. The efficiency interval widens relative to the fixed-policy interval, while the quality interval remains similar; these are uncertainty estimates for a fitted floor, not a quality certificate.

A.4 POLICY ROBUSTNESS: INFERRED STATES AND LEARNED ROUTERS

Protocol. All policies are fitted on probing and evaluated on the test set with 1,000 paired task bootstraps (seed 20260925, so intervals differ slightly from Table 7). Prompts are rebuilt with the benchmark loaders (9,996 of 9,999 tasks); because the loaders add source-specific instruction wrappers, classifier accuracy is an upper bound for raw user prompts. Predicted-state policies assign probing tasks five-fold out-of-fold predictions before refitting action values, so fitting and deployment use the same noisy state. Learned routers predict each model's score from TF-IDF features by ridge regression (out-of-fold R^2 0.37–0.59) and its log cost by ridge regression on the same features (out-of-fold R^2 0.26–0.87), mean-calibrated out of fold; with a constant probing-mean cost the direct router reaches $19.7\times$ at $\Delta V = -0.006$; the direct version maximizes $\hat{v}_m - \lambda \hat{c}_m$, the cascade version also values "primary then Opus" with predicted acceptance, and the budget version maximizes predicted quality under a mean predicted cost. The RouteLLM-style router is a linear TF-IDF classifier of whether Opus beats Qwen3-Max (test AUC 0.921), replacing the original neural encoders. Operating points are chosen on probing.

Classifier errors concentrate in BigCodeBench-derived cells whose prompts are identical across dimensions. The FrugalGPT-style cascade (Chen et al., 2024b) learns, for each model, a TF-IDF scorer of prompt and response (in place of DistilBERT; out-of-fold AUC 0.95–0.98) and searches all 400 ordered chains of at most three models with per-stage thresholds on out-of-fold probing scores; at the $\epsilon \approx 0$ point it chooses Qwen3-Max→Kimi-K2.5→GPT-5.4. Its native budget objective is infeasible at the cell policy's probing cost (0.463 m$), below the cheapest model's 0.532 m$, because every task pays for the first stage. The AutoMix-style cascade (Aggarwal et al., 2024) accepts the small model's answer when a meta-verifier (threshold variant; the POMDP variant gives the same quality) accepts the recorded grok-4.6 judgment, and otherwise escalates; the verifier is an external judge, not self-verification, and live few-shot self-verification calls were not made. The learned

Table 8: Test-set outcomes without oracle state labels and for learned routers. $r = \eta/\eta_B$; cost in m$ per task; compliance is the share of test resamples with $\Delta V \geq -\epsilon$ at $\epsilon = 0$. Intervals use bootstrap seed 20260925, so they differ slightly from those reported for the same policies in Figure 2 and Table 7.

States or router	Operating point	r [95% CI]	Cost	ΔV [95% CI]	Compl.
True 21 cells	$\epsilon = 0$	25.52 [24.26, 26.77]	0.452	-0.0044 [-0.0177, $+0.0077$]	0.22
True 21 cells	$\epsilon = 0$, margin 0.03	18.46 [17.35, 19.61]	0.649	$+0.0125$ [-0.0009, $+0.0258$]	0.97
Predicted 21 cells	unconstrained	28.19 [26.80, 29.50]	0.394	-0.0202 [-0.0342, -0.0065]	0.00
Predicted 21 cells	$\epsilon = 0$	25.32 [24.07, 26.63]	0.453	-0.0070 [-0.0203, $+0.0058$]	0.13
Predicted 21 cells	$\epsilon = 0$, margin 0.03	18.34 [17.23, 19.50]	0.651	$+0.0108$ [-0.0026, $+0.0247$]	0.94
Predicted 9 dims	$\epsilon = 0$	24.40 [23.16, 25.80]	0.472	-0.0048 [-0.0177, $+0.0080$]	0.19
TF-IDF k-means 20	$\epsilon = 0$	25.65 [24.34, 27.00]	0.447	-0.0071 [-0.0207, $+0.0060$]	0.12
TF-IDF k-means 50	$\epsilon = 0$	25.94 [24.62, 27.28]	0.444	-0.0052 [-0.0186, $+0.0072$]	0.19
Pooled (no state)	$\epsilon = 0$	3.62 [3.41, 3.85]	3.232	$+0.0021$ [-0.0085, $+0.0135$]	0.63
Source wrappers removed from every prompt by one rule learned on probing prompts					
Predicted 21 cells	$\epsilon = 0$	24.92 [23.65, 26.28]	0.464	-0.0032 [-0.0163, $+0.0096$]	0.28
Predicted 9 dims	$\epsilon = 0$	23.59 [22.40, 24.88]	0.490	-0.0039 [-0.0169, $+0.0089$]	0.24
TF-IDF k-means 20	$\epsilon = 0$	22.70 [21.45, 23.97]	0.512	-0.0015 [-0.0145, $+0.0109$]	0.39
Quality-cost, direct	probing $\epsilon \approx 0$	21.29 [20.06, 22.57]	0.548	$+0.0003$ [-0.0126, $+0.0131$]	0.50
Quality-cost, direct	probing $\epsilon \approx 0$	24.17 [22.83, 25.49]	0.482	-0.0003 [-0.0140, $+0.0125$]	–
Quality-cost, cascade	probing $\epsilon \approx 0$	22.67 [20.48, 24.72]	0.512	-0.0016 [-0.0142, $+0.0114$]	–
Quality-cost, budget	probing $\epsilon \approx 0$	23.57 [22.21, 24.90]	0.497	$+0.0026$ [-0.0107, $+0.0157$]	–
kNN-20, direct	probing $\epsilon \approx 0$	23.58 [22.28, 24.89]	0.500	$+0.0050$ [-0.0080, $+0.0175$]	–
RouteLLM-style	probing $\epsilon \approx 0$	6.36 [5.82, 6.90]	1.861	$+0.0064$ [-0.0064, $+0.0193$]	–
Quality-cost, cascade	budget = Opus cost	1.76 [1.67, 1.87]	8.037	$+0.0950$ [$+0.0847$, $+0.1049$]	–
FrugalGPT-style cascade	probing $\epsilon \approx 0$	11.76 [11.02, 12.51]	1.000	$+0.0039$ [-0.0093, $+0.0171$]	0.71
FrugalGPT-style cascade	budget = Opus cost	1.84 [1.74, 1.96]	7.400	$+0.0738$ [$+0.0628$, $+0.0844$]	1.00
334 runnable test tasks with recorded judge verdicts; always-Opus scores 0.737 at 5.413 m$					
AutoMix-style, judge free	probing $\epsilon \approx 0$	1.64 [1.47, 1.89]	3.289	-0.0030 [-0.0240, $+0.0150$]	0.43
AutoMix-style, judge at primary price	probing $\epsilon \approx 0$	0.89 [0.83, 0.96]	6.067	$+0.0000$ [-0.0180, $+0.0180$]	0.55
AutoMix-style, judge at list price	probing $\epsilon \approx 0$	0.28 [0.26, 0.31]	19.125	$+0.0000$ [-0.0180, $+0.0180$]	0.55
True 21 cells	$\epsilon = 0$	18.57 [16.53, 20.73]	0.270	-0.0539 [-0.0928, -0.0150]	0.01

routers lie within 0.006 score of the cell-policy frontier at matched cost; the binary router lies about 0.07 below it because it can choose only Opus or Qwen3-Max.

Prompts without source wrappers. The benchmark loaders prepend or append source-specific instructions, which the classifier can read. A line or sentence counts as a wrapper template if it occurs in at least 20% of one source's probing prompts, after masking inline code spans; one rule removes every template of every source, and the longest matching common prefix and suffix, from every prompt, so it needs no source label when applied. It removes 19% of test-prompt characters and leaves a template phrase in 4.5% of test prompts, against 88% before. Cell accuracy falls from 92.8% to 76.6%, mostly between LiveCodeBench's algorithm and test-generation cells, which pose the same problems; the benchmark source itself is still recognized in 96.8% of test prompts, from task content alone. The lower block of Table 8 refits every state scheme and the learned router on the stripped prompts.

In the first 200 policy-refitting bootstrap replicates above, we also refit the stripping rule, TF-IDF features, and cell classifier before refitting the policy. Out-of-fold predictions group duplicate copies of the same task in one fold. The predicted-cell $\epsilon = 0$ policy has median efficiency $24.83\times$ $[22.09, 27.38]$ and median $\Delta V = -0.0026$ $[-0.0163, +0.0115]$. This supports within-distribution routing from prompt text without source wrappers or benchmark labels at deployment; it does not establish transfer to unseen sources.

A stronger yardstick. Opus has the highest mean probing score of the eight models, $+0.020$ $[0.012, 0.028]$ above GPT-5.4, and the highest test score. A stronger yardstick routes each cell (CellBest) or dimension (DimBest) to its best probing model. Both are routers over the menu, hence admissible; we refit the floors against each yardstick B', with cascades still falling back to Opus, and report $r' = \eta/\eta_{B'}$ and $\Delta V' = \mathbb{E}[V - Y_{B'}]$ on the test set (Table 9). Against either yardstick the refitted zero-loss policy is not significantly worse and is 2.3–3.2 times more efficient, so the qualitative result survives. The size of the multiplier does not: most of the $25.5\times$ reflects Opus's price, and the Opus-referenced policy falls 0.083–0.087 below both yardsticks. A floor certifies quality only relative to the reference it names.

Table 9: Zero-loss floors against stronger yardsticks on the test set. r' and $\Delta V'$ are relative to the yardstick, r to Opus; cost in m$ per task; compliance is the share of resamples with $\Delta V' \geq 0$.

Yardstick	Policy	r' [95% CI]	r	Cost	$\Delta V'$ [95% CI]	Compl.
CellBest	itself	1	2.32	5.969	0 (+0.083 vs. Opus)	–
	21 cells, $\epsilon = 0$	3.18 [2.94, 3.40]	7.37	1.865	−0.0039 [−0.0108, +0.0027]	0.13
	9 dims, $\epsilon = 0$	2.34 [2.18, 2.52]	5.43	2.536	−0.0034 [−0.0104, +0.0038]	0.18
	Opus-referenced 21 cells, $\epsilon = 0$	11.00 [10.46, 11.54]	25.52	0.452	−0.0870 [−0.0978, −0.0768]	0.00
DimBest	itself	1	3.07	4.483	0 (+0.079 vs. Opus)	–
	21 cells, $\epsilon = 0$	2.68 [2.49, 2.90]	8.20	1.663	−0.0037 [−0.0092, +0.0018]	0.09
	Opus-referenced 21 cells, $\epsilon = 0$	8.32 [7.83, 8.86]	25.52	0.452	−0.0829 [−0.0926, −0.0736]	0.00

Ratio, price, and budget. On the true-cell menu, the $\epsilon = 0$ ratio policy has multiplier $\kappa = 5.43$ and constrained optimum $\eta^\star_\epsilon = 927.7$ score per dollar, so $\lambda_\epsilon = \eta^\star_\epsilon/(1+\kappa) = 144.3 = 4.16\,\eta_B$. It attains $\max_g J_{\lambda_\epsilon}$ exactly, equals the budget-constrained quality maximizer at its own cost (budget dual 144.3), and equals the minimum-cost policy subject to the floor. On held-out data it lies within 0.001 score of the test J_λ hull at its cost. The ratio view thus removes the choice of λ. Defaults that do not use the floor land elsewhere: $\lambda = \eta_B$ gives $10.85\times$ at 1.24 m$ and $\Delta V = +0.0665$ $[+0.0544, +0.0786]$, while a budget equal to Opus's cost is slack and yields pure quality maximization, $2.16\times$ at 6.68 m$ and $+0.1050$. Choosing the margin by plain probing bootstrap gives 0.015–0.0175 and test compliance 0.47–0.96, whereas requiring a one-sided 95% lower bound on probing gives 0.0275–0.03 and compliance 0.94–0.999.

A.5 Fixed-menu policies on RouterBench

We repeat the study of Section 3.1 on RouterBench with gpt-4-1106-preview as the reference; it scores 0.781 at 3.29 m$ per held-out prompt. Fitting and held-out prompts are a 70/30 split stratified by dataset and reference success (25,542 and 10,955 prompts), built like the candidate-selection splits. The menu has 21 actions: direct dispatch to each of the eleven models, and each non-reference model followed by the reference when its released score is below $\theta = 1$ (oracle acceptance, no verifier fee). States are the eight source families, the 86 datasets (57 MMLU subsets), datasets predicted from the prompt by TF-IDF logistic regression (92.0% held-out accuracy, 79.8% within MMLU), TF-IDF k-means clusters, or one pooled state. Learned routers predict each model's score, log cost, and acceptance by kernel ridge regression on TF-IDF features, or average the outcomes of the 20 nearest fitting prompts; each runs at the price or budget that is closest to zero loss on fitting prompts. Intervals use 1,000 paired resamples of held-out prompts within dataset, with policies fixed.

Table 10: Fixed-menu policies on 10,955 held-out RouterBench prompts at $\theta = 1$. $r = \eta/\eta_B$ against gpt-4-1106-preview; cost in m$ per prompt; compliance is the share of resamples with $\Delta V \geq 0$.

States or router	Operating point	r [95% CI]	Cost	ΔV [95% CI]	Compl.
Cheapest model (Mistral-7B)	–	28.56 [27.78, 29.28]	0.045	−0.4726 [−0.4825, −0.4635]	0.00
Best fixed cascade (Yi-34B, then reference)	probing	1.38 [1.37, 1.39]	2.515	+0.0425 [+0.0390, +0.0461]	1.00
8 families	unconstrained	32.11 [31.37, 32.81]	0.052	−0.3817 [−0.3910, −0.3722]	0.00
8 families	$\epsilon = 0$	7.10 [6.91, 7.33]	0.461	−0.0040 [−0.0084, +0.0008]	0.05
86 datasets	$\epsilon = 0$	8.26 [8.01, 8.51]	0.395	−0.0073 [−0.0125, −0.0024]	0.00
Predicted 86 datasets	$\epsilon = 0$	7.85 [7.55, 8.16]	0.416	−0.0057 [−0.0112, −0.0009]	0.01
TF-IDF k-means 20	$\epsilon = 0$	8.10 [7.82, 8.39]	0.405	−0.0024 [−0.0079, +0.0026]	0.17
Pooled (no state)	$\epsilon = 0$	1.71 [1.70, 1.73]	1.910	−0.0035 [−0.0073, +0.0008]	0.06
Quality-cost, cascade	price, fit $\epsilon \approx 0$	8.10 [7.89, 8.32]	0.405	−0.0025 [−0.0080, +0.0029]	0.18
Quality-cost, cascade	budget, fit $\epsilon \approx 0$	7.07 [6.88, 7.27]	0.474	+0.0152 [+0.0104, +0.0199]	1.00
kNN-20, cascade	price, fit $\epsilon \approx 0$	6.71 [6.50, 6.94]	0.492	+0.0026 [−0.0024, +0.0076]	0.83
Quality-cost, direct	price, fit $\epsilon \approx 0$	1.65 [1.62, 1.68]	1.998	+0.0015 [−0.0015, +0.0044]	0.84
RouteLLM-style, reference vs. Mixtral	fit $\epsilon \approx 0$	1.00	3.289	0	1.00
Per-task oracle (in-sample)	$\epsilon = 0$	45.29 [44.68, 45.87]	0.073	0.0000 [−0.0073, +0.0072]	0.48

The price structure is the opposite of CodeRouterBench's. With family states, no reference-free router comes within 0.097 of the reference ($\delta^{\text{free}} = -0.097$), and the admission price $\lambda^B = 5.85\eta_B$ exceeds the zero-loss floor price $1.60\eta_B$, so the reference is indispensable in the sense of Proposition 4(b)(ii): the $\epsilon = 0$ policy sends 68% of prompts to cascades that call it, and it runs on 21%. The same holds for every state scheme, and the band bound of part (a) holds on all 81 admitted state–cascade pairs. Learned direct-dispatch routers, which cannot call the reference only after a failure, reach at most

$1.65\times$ at their zero-loss points. Fitted floors again miss out of sample: on the original and the ten candidate-selection splits, the family $\epsilon = 0$ policy keeps held-out $\Delta V \geq 0$ in 4 of 11 and the dataset policy in none. Certified on a random half of the fitting prompts (Proposition 3, $\alpha = 0.05$), the family floor keeps held-out $\Delta V \geq 0$ in all 200 halvings at a median $6.70\times$ with empirical-Bernstein and $5.97\times$ with Hoeffding p-values, and the dataset floor at $6.80\times$, whereas the uncertified floor refitted on the same halves holds in none. At matched cost, the learned quality-cost cascades lie $+0.014$ $[0.010, 0.018]$ (budget) and $+0.022$ $[0.017, 0.026]$ (price) above the 8-family frontier and tie with the 86-dataset one (-0.002 $[-0.006, +0.002]$ and $+0.001$ $[-0.004, +0.005]$); kNN cascades lie 0.019 below the latter. Prompt-level quality prediction thus adds value where states are coarse, and the floor again selects an operating point rather than a better frontier.

A.6 Failure-conditioned chains and online retries

Recorded chains. The chain-planning study enumerates 259 ordered chains of one to three distinct non-Opus models, followed by Opus, at $\theta \in \{1, 0.5\}$. A post-failure state contains the identities of all models already tried and their failures. Probing estimates determine whether to attempt another model or stop and call Opus; the selected chain is then replayed on test tasks. A segment is one continuation decision, indexed by cell, chain, stage, and price; it is interval-decided when the paired interval of its realized gain excludes zero, and degenerate when the attempted model never succeeded in that state on probing. The prices are $\lambda \in \{0.1, 1, 10, \eta_B, 100\}$ score per dollar, with $\eta_B = 34.727$ on probing. At $\theta = 1$ they yield 57,405 interval-decided 21-cell segments, shared across tasks and chains. Sign agreement is 88.5% for the shared-value rule of Proposition 10, 77.2% for marginal planning, 93.2% for the state-matched recursion, and 67.5% for never attempting. Restricting to 34,746 non-degenerate segments gives 81.2%, 69.6%, and 47.2% for the shared-value, marginal, and never-attempt rules, respectively. At $\theta = 0.5$, the shared-value and marginal rules agree with 87.6% and 77.8% of decided signs.

Single-step control. The one-step decision compares an attempted model followed by Opus against calling Opus directly, using two primaries, two thresholds, and five resource prices. Primitives are fitted on probing. Across the 9-dimension and 21-cell partitions, the value-cost rule agrees with 92.5% and 93.3% of interval-decided signs (148/160 and 305/327, from 180 and 420 total). The cost-only threshold $p \geq a/b$ achieves 95.0% and 92.0%, versus 76.3% and 63.9% for always attempting. On all comparisons, value-cost versus cost-only agreement is 88.3% versus 90.6% and 91.0% versus 90.2%. At the probing reference price $\lambda = \eta_B = 34.727$, the two rules make identical decisions, which agree with 100% of the 36 decided dimension signs and 94.0% of the 84 decided cell signs. Mean loss relative to the better held-out action at this price is respectively 0 and 0.000657 score units, versus 0.003276 and 0.006933 for always attempting.

Stage-dependent costs. The retry state contains the previous answer and at most 800 characters of test feedback. Token and dollar charges are measured for each reached stage, including repeated input context. For Qwen3-Max, mean tokens rise from 523 at the first attempt to 1,276 at the first feedback retry and 2,101 at the second; corresponding costs rise from 0.49 to 0.87 and 1.25 m$ per reached attempt. Qwen3.5-Plus uses 536, 1,309, and 2,398 tokens, costing 0.22, 0.41, and 0.67 m$. These means refer to different surviving task subsets. They enter $a(\mathfrak{h}, u)$ in Eq. (9); the planner does not reuse the first call's charge at later stages.

Online retries with a recorded terminal. The online retry study makes 1,415 first-attempt chains on 283 executable test tasks, excluding tasks scored zero by all eight recorded models. Five models (request identifiers `qwen3-max`, `kimi-k2.5`, `qwen3.5-plus`, `deepseek-v4-pro`, and `deepseek-flash`) are called through one OpenAI-compatible API relay at temperature 0.7 with completion budgets of 2,048 tokens, or 6,144 for `deepseek-flash`; exhausted attempts count as failures. The relay does not report a served model version, so identities are the requested ones. After failure, we compare a fresh retry, a retry receiving failing-test output, and a model switch receiving the same context. Opus is the recorded terminal. The hidden-test sandbox both supplies retry feedback and scores the output, so improvement measures repair against those tests.

Because the same tests supply the feedback and score the retry, this protocol measures repair against tests the retry has seen. The main text therefore reports only the rerun below, which keeps feedback and scoring tests disjoint.

Disjoint feedback and scoring tests. Each task's test cases are split with a fixed seed into a feedback half F and a disjoint scoring half S, at the level of individual assertions or test methods. One task with a single separable test is ineligible, leaving 282 tasks. A first attempt fails if it fails S. We expand the study to seven models and three samples per retry arm: the five original models reuse their archived first attempts and original fresh/feedback retry pair, with two additional pairs; GLM-5 and MiniMax-M2.7 each add first attempts on the 282 tasks and three pairs per S-failure. The extension uses the same F/S split, relay, and temperature 0.7. Completion budgets remain 2,048 tokens for the original models except DeepSeek-flash (6,144); the two added models use 4,096. Feedback contains at most 800 characters from F only; if the first attempt passes F, it states only that all provided tests pass. Every retry is scored on S.

The expanded evaluation contains 263 failed task–model pairs across 88 tasks and 1,578 retry outcomes. The extension adds 564 first-attempt records and 1,216 distinct retry records, including cached responses. Recovery is the mean per-retry success rate: we first average the three samples within each failure and arm, then average across failures. It is not the probability that at least one of three retries succeeds. Paired 95% percentile intervals use 2,000 task-cluster bootstrap resamples (seed 20260926), keeping all models and samples from a resampled task together.

Model	Failures	Fresh	Feedback	Difference [95% CI]
DeepSeek-flash	52	0.263	0.327	$+0.064$ $[-0.050, +0.180]$
DeepSeek-v4-pro	34	0.157	0.363	$+0.206$ $[+0.061, +0.365]$
Kimi-K2.5	35	0.105	0.219	$+0.114$ $[-0.010, +0.241]$
Qwen3-Max	31	0.032	0.280	$+0.247$ $[+0.102, +0.407]$
Qwen3.5-Plus	29	0.069	0.218	$+0.149$ $[-0.021, +0.333]$
GLM-5	28	0.071	0.262	$+0.190$ $[+0.051, +0.358]$
MiniMax-M2.7	54	0.290	0.383	$+0.093$ $[-0.012, +0.194]$
Pooled	263	0.165	0.304	$+0.139$ $[+0.058, +0.223]$

All seven model-level point estimates are positive. In the 52 failures whose first attempt passes every feedback test, feedback contains no failing output and reduces recovery: 0.064 versus 0.173 fresh, a difference of -0.109 $[-0.217, -0.025]$. In the other 211 failures, the difference is $+0.201$ $[+0.109, +0.290]$. Excluding 59 truncated first attempts (33 DeepSeek-flash and 26 MiniMax-M2.7) leaves 204 failures, with recovery 0.291 feedback versus 0.126 fresh and difference $+0.165$ $[+0.066, +0.271]$.

A.7 Verifier operating points

The verifier study evaluates recorded responses from three primaries on 500 source-stratified probing and 334 runnable test tasks. Compiler is an execution-free static gate: extraction, AST/bytecode compilation of the candidate and official harness, and a declared-entry-point check; it never imports or executes code. The other verifiers are visible examples, hidden tests, the released score oracle, and a Grok-4.6 judge (Qwen3-Max and Kimi-K2.5 only; 1,668 online calls). Table 11 charges the judge at the primary's blended token price, a proxy. At xAI's list price for grok-4.6 ($2 and $6 per million input and output tokens), the mean judgment costs $0.0159 for Qwen3-Max and $0.0143 for Kimi-K2.5, 4.8–5.8 times the $\theta = 0.5$ break-even charge of Eq. (12); the efficiency ratios become 0.297 and 0.251 (Qwen3-Max, $\theta = 0.5$ and 1) and 0.303 and 0.270 (Kimi-K2.5). Sandbox executions are charged at their measured price (Appendix A.9).

For Qwen3-Max, Compiler has AUROC 0.766, $r = 4.09$ at both thresholds, and $Q = -0.024$ $[-0.063, +0.009]$ at $\theta = 0.5$. Accepting every output reaches 19.65-fold efficiency but loses $Q = -0.063$ $[-0.108, -0.024]$, so the quality floor excludes that tradeoff when the loss is unacceptable. The operating-point floor predicts 97.6% of 416 decided signs among 450 operating points, and the predicted minimum quality-feasible threshold lies in the realized interval in 95.0% of 360 cells.

Absolute price boundaries from archived tokens. Let $\bar{D}_{0,\theta}$ be mean primary-plus-fallback cost before the judge's charge, and $\bar{V}_\theta$ the resulting mean delivered score. Holding outputs and threshold fixed, the judge improves efficiency precisely when its mean dollar charge satisfies

$$\bar{c}_{\text{judge}} < \frac{\bar{V}_\theta}{\eta_B} - \bar{D}_{0,\theta}. \tag{12}$$

Table 11: Verifier choice changes the cascade's efficiency and quality. $r = \eta/\eta_B$; Q is mean score change. Judge rows use the primary-price scenario; Local verification time enters seconds.

Verifier	AUROC	$r(\theta = 0.5)$ [95% CI]	$r(\theta = 1)$ [95% CI]	$Q(\theta = 0.5)$
Compiler (static gate)	0.766	4.09 [3.23, 5.45]	4.09 [3.23, 5.45]	-0.024
Visible examples	0.537	1.98 [1.72, 2.35]	1.06 [1.03, 1.10]	-0.036
Hidden tests	0.994	2.87 [2.38, 3.53]	2.37 [2.03, 2.80]	$+0.024$
LLM judge	0.824	1.09 [1.00, 1.21]	0.64 [0.61, 0.66]	-0.012
Released-score oracle	1.000	2.46 [2.09, 2.93]	2.46 [2.09, 2.93]	$+0.045$

The right side is the break-even charge, independently of how a vendor prices input and output. At $\theta = 0.5$, the boundary is \$0.003300 $[0.002849, 0.003786]$ per task for Qwen3-Max and \$0.002470 $[0.002028, 0.002974]$ for Kimi-K2.5. At $\theta = 1$, the respective boundaries are $-\$0.000247$ $[-0.000256, -0.000234]$ and $-\$0.000352$ $[-0.000389, -0.000318]$: even a free judge does not improve efficiency at that operating point. Intervals use 2,000 paired task-bootstrap resamples of the 334 held-out judgments per primary, recomputing reference efficiency in each resample while fixing outputs and thresholds. At $\theta = 0.5$, mean quality changes remain -0.012 and -0.039, respectively, and must satisfy the chosen quality floor separately.

The Qwen3-Max judgments average 1,302.25 input and 2,218.44 output tokens; Kimi-K2.5 averages 1,298.01 input and 1,945.51 output tokens. Equal input and output prices yield positive boundaries of \$0.937 and \$0.762 per million tokens; unequal prices are weighted by these token counts.

An additional sensitivity multiplies the original primary-price charge by t. With $r(0) = 2.63$ and $r(1) = 1.09$ for Qwen3-Max at $\theta = 0.5$, the resulting efficiency ratio is $r(t) = [r(0)^{-1} + t\{r(1)^{-1} - r(0)^{-1}\}]^{-1}$. Halving that charge gives 1.54, doubling it gives 0.69, and break-even occurs near $t = 1.16$. Together with Eq. (12), this gives the price range in which the judge helps.

A.8 Deployable verifiers inside the policy menu

The fixed-menu study of Section 3.1 accepts a cascade's primary on its released score, for free. To see what a deployable verifier changes, we rebuild that study on the verifier sample of Appendix A.7: policies are fitted on the 489 probing tasks that both verifier runs share and evaluated on the 334 test tasks. Every menu contains direct dispatch to the eight models. The oracle menu adds, for the three primaries and $\theta \in \{1, 0.5\}$, released-score cascades at no verifier charge; the deployable menu instead adds cascades that accept on the recorded Compiler, visible-test, hidden-test, or judge verdict and pay the measured charge of Appendix A.9 (the judge at list price or at the primary-price proxy). A false accept delivers the primary's score, and a rejection falls back to Opus's recorded outcome. Only five of the nine dimensions occur in this sample, so states are the dimensions or a single pooled state; floors use Opus's fitted value, and intervals use 2,000 dimension-stratified task resamples with policies fixed.

Table 12: Zero-loss floor policies ($\epsilon = 0$) with oracle or deployable acceptance on the 334-task verifier sample. $r = \eta/\eta_B$ against always-Opus (0.737 at 5.413 m\$ per task); cost in m\$ per task; compliance is the share of resamples with $\Delta V \geq 0$. Cascades: share of test tasks sent to a verified cascade and its verifier.

Menu	States	r [95% CI]	Cost	ΔV [95% CI]	Compl.	Cascades
Direct dispatch only	pooled	1.00	5.413	0	1.00	none (always Opus)
	5 dims	2.04 [1.94, 2.15]	2.516	-0.0377 $[-0.0672, -0.0093]$	0.01	none
Oracle acceptance, free	pooled	3.72 [3.22, 4.39]	1.452	-0.0008 $[-0.0268, +0.0269]$	0.45	58%, released score
	5 dims	5.93 [5.03, 7.04]	0.871	-0.0335 $[-0.0695, +0.0006]$	0.03	53%, released score
Deployable, measured charges	pooled	3.60 [3.13, 4.18]	1.500	-0.0025 $[-0.0280, +0.0240]$	0.40	58%, hidden tests
	5 dims	4.89 [4.14, 5.87]	1.066	-0.0269 $[-0.0612, +0.0065]$	0.06	57%, hidden tests
Deployable without hidden tests	pooled	2.33 [2.03, 2.71]	2.277	-0.0153 $[-0.0447, +0.0146]$	0.16	100%, Compiler
	5 dims	2.74 [2.34, 3.31]	1.912	-0.0233 $[-0.0612, +0.0123]$	0.10	57%, Compiler
Unconstrained ratio optimum, every menu	both	19.65 [17.62, 21.81]	0.252	-0.0629 $[-0.1048, -0.0210]$	0.00	none (Qwen3-Max)

Table 12 gives three answers. First, the verifier does not change the unconstrained ratio optimum, which dispatches every task to Qwen3-Max in every menu; verifiers matter only through the floor. Second, under the floor the deployable menu replaces each oracle cascade by a hidden-test cascade, with the same primary except in code generation, where Kimi-K2.5 replaces Qwen3-Max; it keeps

97% of the oracle menu's pooled efficiency (paired difference -0.13 $[-0.29, -0.02]$) and 82% with dimension states (-1.04 $[-1.68, -0.58]$). Hidden tests are the benchmark's own grader, so the row without them is the deployable one: the fitted floor then uses only Compiler cascades and reaches $2.33\times$ pooled, 1.33 $[1.03, 1.71]$ above direct dispatch, which meets the pooled floor only by always calling Opus. Third, the judge enters no fitted policy at its list price ($0.0159 per judgment, 2.9 Opus calls) or at the proxy ($0.0029), visible examples enter at most 4.5% of 200 probing refits of the full deployable menu, and a free judge would change r by at most 0.004. In sequential seconds the dollar-fitted floor policies reach 0.50–0.55$\times$ Opus with hidden-test or oracle cascades.

With five dimension states, no fitted floor holds out of sample in any menu, direct dispatch included: test ΔV lies between -0.038 and -0.023, and 0–3% of 200 probing refits reach $\Delta V \geq 0$. The cause is estimation, not verification: Opus scores 0.013 higher on these test tasks than on probing, and the fitted policies lose on test what they gain on probing, as expected from few tasks per state. The pooled floors, fitted on 489 tasks, come within 0.003 of zero with hidden-test or oracle cascades.

A.9 Measured router, verifier, and latency costs

Replay excludes router and verifier charges; Table 13 measures them. Router inference refits the prompt-only 21-cell classifier and the ridge quality-cost predictors of Appendix A.4 on the probing prompts and times them on the 2,919 test prompts on one Apple M4 core, batched and one prompt at a time. Sandbox and judge times are those recorded by the verifier study on its 334 test tasks; judge calls ran four to ten at a time on a shared relay. CPU time is priced at AWS Lambda's x86 list price for a 2 GB function, $\$3.33 \times 10^{-5}$ per second plus $\$2 \times 10^{-7}$ per request (us-east-1, September 2026).

Table 13: Measured per-task overheads. Ratios compare with the mean Qwen3-Max call on the same tasks ($0.000525 on the test set, $0.000252 and 6.23 s on the 334-task verifier sample); Opus costs $0.00541 and takes 3.93 s there.

Component	Time per task	Dollars per task	vs. Qwen3-Max
TF-IDF features, 21-cell classifier, 16 ridge predictors	1.9 ms median, 3.6 ms p95	3.2×10^{-7}	0.06%
kNN-20 over probing prompts (batched)	1.3 ms	2.4×10^{-7}	0.05%
Visible tests, one sandbox run on 50% of tasks	0.16 s mean	5.6×10^{-6}	2.2%
Hidden tests, one sandbox run	0.15 s mean, 0.08 s median	5.2×10^{-6}	2.0%
LLM judge (grok-4.6), 3,521 tokens	39.0 s mean, 28.3 s median	0.0159 at list price	63$\times$
LLM forecast (DeepSeek-v4-pro), 635 tokens	1.89 s mean	0.000944 at peak list price	2.4$\times$

Charging the measured sandbox time changes dollar efficiency by at most 0.01 (Table 14). The break-even mean charge of Eq. (12) is 209–705 times the measured sandbox charge at $\theta = 0.5$ across verifiers and primaries. Router overhead is below 0.1% of a primary call, so charging it leaves every replay comparison unchanged at the reported precision. In sequential seconds, Opus answers faster than Qwen3-Max and Kimi-K2.5 on these tasks, so direct dispatch reaches $0.58\times$ and $0.50\times$ Opus efficiency and every verified cascade stays below one; sandbox time lowers seconds efficiency by at most 0.011, whereas the judge's call latency lowers it to 0.08–$0.09\times$. The fixed-menu policies of Section 3.1 reach 0.71–$0.73\times$ in seconds. Router inference, at a few milliseconds, is negligible against call latencies of seconds.

Fitting directly for mean seconds. We also refit the 21-cell, 22-action menu of Section 3.1 on probing with sequential seconds as the resource. We replace recorded zero latencies by the same model and cell's median positive probing latency. The $\epsilon = 0$ policy reaches $1.22\times$ Opus's quality-per-second efficiency $[1.18, 1.26]$, with $\Delta V = +0.0132$ $[+0.0063, +0.0205]$ and mean latency 6.27 s versus Opus's 7.41 s. Intervals use 1,000 paired test-task bootstrap resamples with the policy fixed. Keeping the released zero latencies gives the same rounded efficiency and quality results. Thus choosing seconds before fitting changes the policy and its performance, compared with evaluating a dollar-fitted policy in seconds.

Table 14: Effect of measured verifier costs at $\theta = 0.5$ on the 334-task verifier sample. $r_{\$}$: dollar efficiency ratio with a free verifier and with the measured sandbox charge. r_{sec}: sequential-seconds ratio without and with verifier time. Break-even: the charge of Eq. (12) divided by the measured sandbox charge.

Primary	Verifier	$r_{\$}$ free	$r_{\$}$ measured	r_{sec} without	r_{sec} with	Break-even
Qwen3-Max	visible tests	1.98	1.98	0.486	0.476	458×
	hidden tests	2.87	2.86	0.539	0.529	705×
	LLM judge	2.63	–	0.526	0.083	–
	released score	2.46	2.46	0.537	0.537	–
Kimi-K2.5	visible tests	1.56	1.56	0.463	0.453	312×
	hidden tests	1.71	1.70	0.510	0.501	448×
	LLM judge	1.93	–	0.521	0.093	–
Qwen3.5-Plus	visible tests	1.30	1.30	0.092	0.092	209×
	hidden tests	1.68	1.68	0.098	0.098	427×

Tail latency as a separate objective. For p95 sequential seconds, we select on probing from 100 distinct candidates comprising deterministic price-path policies for mean seconds and dollars (21 cells and pooled) and the 22 fixed actions. Maximizing mean quality divided by p95 latency subject to the fitted zero-loss floor gives $1.18\times$ Opus's score/p95 efficiency $[1.13, 1.22]$, with test p95 20.62 s versus Opus's 23.55 s under the same imputation. This is selection over the stated candidate family; p95 is not an expectation, so the mean-cost optimization theorem does not apply to this objective.

A.10 Released and online routing signals

Released signals. The nine CodeRouterBench router files expose the chosen model, not a full forecast. The released-signal analysis therefore forms binary signals: chose Qwen3-Max or chose any non-Opus model. Primitive costs and base rates come from probing, while posterior lifts are cross-fitted over ten half-splits of the test state. Native use attempts on $Z = 1$; posterior-threshold use attempts when $s_Z > a/b$. Capture is gross cost saving divided by the perfect-information value, before the router's own charge. The pooled baseline receives no task-type label; the dimension-conditioned diagnostic receives the benchmark's task type.

For $Z =$ chose Qwen3-Max, classical routers have negative native capture because their $Z = 0$ tasks still satisfy $\mathbb{P}(H = 1 \mid Z = 0) > a/b = 0.043$: withholding the attempt loses profitable opportunities. Posterior-threshold use of those signals has zero capture. Defining Z as choosing any non-Opus model changes the question and gives the classical routers 0.03–0.22 threshold capture, still below the two largest fine-tuned routers.

Table 15: Cross-fitted capture of all nine released CodeRouterBench routers as binary signals for Qwen3-Max at $\theta = 1$, with no signal charge. Native use attempts on $Z = 1$; threshold use attempts when $s_Z > a/b$. The within-dimension column uses the threshold rule with the benchmark dimension supplied as oracle metadata and divides aggregated gross saving by the aggregated perfect-information bound. The last two columns redefine Z as choosing any non-Opus model.

	$Z =$ chose Qwen3-Max			$Z =$ chose a non-Opus model	
Signal source	Native [95% CI]	Threshold [95% CI]	Within dim.	Native	Threshold
Fine-tuned Qwen 27B	0.555 [0.490, 0.614]	0.555 [0.490, 0.614]	0.062	0.337	0.337
Fine-tuned Qwen 9B	0.546 [0.475, 0.608]	0.546 [0.475, 0.608]	0.078	0.325	0.325
Fine-tuned Qwen 2B	0.466 [0.397, 0.531]	0.417 [0.355, 0.479]	−0.006	0.339	0.132
Fine-tuned Qwen 0.8B	0.422 [0.348, 0.491]	0.082 [0.066, 0.098]	−0.008	0.296	0
Logistic regression	−1.218 [−1.590, −0.869]	0	−0.002	0.206	0.206
TF-IDF MLP	−1.736 [−2.178, −1.342]	0	−0.012	0.162	0.125
RouteLLM MF	−1.033 [−1.364, −0.723]	0	−0.013	0.217	0.217
RouteLLM SW	−1.284 [−1.685, −0.953]	0	+0.005	0.136	0.033
DimensionBest	−1.468 [−1.847, −1.104]	0	0	−0.843	0

A fresh LLM forecast. The paid-forecast study samples 600 probing and 600 test tasks by source and cell from 3,601 executable, non-all-zero tasks. DeepSeek-v4-pro reads the prompt and predicts whether Qwen3-Max and Kimi-K2.5 pass all hidden tests (1,200 calls). Its posterior is fitted by pooled

probing isotonic regression and anchored to the state's probing base rate; the decision threshold is a/b. Held-out AUROC is 0.541 and 0.618, versus 0.895 and 0.768 for TF-IDF logistic routing fitted on the same prompts. The forecast's task-weighted capture across 12 non-degenerate source×primary rows is -0.037 $[-0.225, 0.077]$.

At the selected peak list price, a call averages 635 tokens, 1.89 seconds, and \$0.000944. It costs 2.4 Qwen3-Max attempts as a ratio of task means, or 2.87 when averaging state-wise ratios, and exceeds the attempt cost a in 22 of 24 source×primary states, so the necessary condition $k < a$ of Proposition 9 fails there before accuracy enters.

A.11 Router memory

Design. The memory study combines real CodeRouterBench outputs and verifier judgments with simulated record arrivals and retention. The main execution population contains 2,044 probing and 867 test tasks in 12 dimension×source cells; tasks scored zero by every recorded model in the affected execution sources are excluded. Variants use 21 cells or 50 TF-IDF k-means clusters fitted on probing. On each round, the fixed writer explores a uniformly sampled model with probability $\varepsilon = 0.25$ and writes its identity to the arriving cell if verification accepts. The newest retained model serves subsequent tasks in that cell; uncovered cells use always-Opus or a DimensionBest prior fitted on probing. Arrivals, exploration, and expiration are constructed because the release contains no memory-write log.

Writers use ACRouter's doctest/syntax tools, locally executed hidden tests, or a DeepSeek-v4-pro judge on a subsample. The 401 empty responses among 6,936 main-population task-model pairs are assigned their released acceptance truth and excluded from verifier error-rate estimates. Fifty streams of 60,000 task draws use the final 10,000 rounds for stationary measurements. We compare geometric expiration, TTL, and event-FIFO retention. Probing estimates predict the gain before the test streams are evaluated; intervals resample streams and, in a separate analysis, task clusters.

Findings. The same store gains against Opus and not against an already efficient prior, as the criterion predicts. In the 12-cell hidden-test setting without expiration, where all cells become covered, stationary efficiency is 135.8 score per dollar. The store's mean score is 0.034 below always-Opus and 0.116 below DimensionBest; the gains in Table 16 price these quality changes against the saving at each prior's efficiency. The record criterion correctly predicts opposite gains against the two priors (Table 16). At the cell level it predicts 918/969 interval-decided signs (94.7%), compared with 646/969 (66.7%) for always predicting a positive gain.

Table 16: The same hidden-test-written store compared with two priors. Gain is the expected per-round surplus $\mathbb{E}[V - \eta_U D]$ at each prior's own efficiency. Brackets are paired 95% intervals.

Prior	Probing-predicted gain	Realized gain [CI]
Always-Opus	$+0.304$	$+0.275$ [$+0.272$, $+0.279$]
Probing-fitted DimensionBest	-0.148	-0.120 [-0.126, -0.113]

Writer disagreement with the released scores is substantial. Against released-score truth, doctest/syntax TPR/FPR is 0.827/0.539 and hidden-test TPR/FPR is 0.960/0.205 on 6,535 non-empty response pairs; the judge gives 0.776/0.341 on 2,320 subsample pairs. The hidden-test FPR of 0.205 differs from the AUROC of 0.994 in Appendix A.7 because of population and truth definition, not a separate harness: the two caches agree on 1,000 of 1,002 shared task–model verdicts. The memory population adds CRUXEval, DS-1000, and LiveCodeBench, which supply 673 of the 685 false accepts (source FPR 0.344, 0.328, and 0.134, versus 0.004 on MBPP). On the verifier study's 334 tasks, the same writer has TPR/FPR 0.993/0.013 and AUROC 0.990. Moreover, 358 false accepts are CRUXEval responses with released score 0.8 whose prediction passes the official assertion; counting them as passes lowers FPR to 0.122. The remaining DS-1000 and LiveCodeBench disagreements are unresolved and may include released-score errors, so these rates bound disagreement with the released labels rather than verifier error alone. For event FIFO, using realized write rates gives mean absolute coverage error 0.0064 for the exact retention expression and 0.0085 for replacing the random span by its mean. Using probing-estimated write rates instead gives held-out errors 0.0495

and 0.0507; uncertainty in write rates is therefore much larger than the correction to the retention approximation.

Error injection and the record threshold. In the main setting no cell lies above its threshold, so we inject writer errors on common random numbers while keeping outputs, verdicts, arrivals, and exploration fixed (semi-synthetic: the injection coins and, in the unverified-serving variant, the record values are constructed; $\rho = 0$ reproduces the stream above exactly). Accepting hidden-test-rejected, released-incorrect outputs with probability ρ raises realized e_j up to 0.90. The aggregate gain keeps its sign (always-Opus $+0.275 \to +0.247$; DimensionBest $-0.120 \to -0.149$) because injected erroneous identities mostly name cheaper, still-usable models, so g_j^W turns positive and the premise $g_j^I > 0 > g_j^W$ of Eq. (11) holds on probing primitives in at most one cell once errors are injected. One decided cell sign changes (always-Opus CRUXEval, $+0.004 \to -0.021$); there $g_j^I < 0$ and injection drives g_j^W negative. Probing-based cell-sign predictions agree with 720 of 721 decided (cell, ρ) signs. Across three mechanisms (false accepts, thinned informative writes, symmetric verdict flips), identity-record gains change sign in five (mechanism, prior, cell) combinations; Eq. (11) with the test population's primitives anticipates all five and with probing primitives one, e.g. always-Opus code completion (HumanEval), which crosses at $e = 0.371$ $[0.328, 0.449]$ against a probing threshold of 0.39–0.40. When stored answers are served without verification (Proposition 12), probing-estimated thresholds locate the 27 decided sign flips with median error 0.11 in e over 15 cells, and anticipate 19. The threshold therefore predicts where a cell's record turns harmful when the premise holds, while its location is limited by how well probing estimates the record-state gains.

Gain signs and magnitude estimation. For 46 geometric-retention settings, we also predict cumulative gain from an empty store over 60,000 rounds. All 46 probing-based predictions agree with the realized sign, but only two (4.3%) fall within the 95% stream-bootstrap interval for mean cumulative gain. Median absolute error, normalized by $T \sum_j \nu_j |G_j|$ at horizon T, is 7.7%. The stream-bootstrap intervals omit uncertainty from estimating the primitives on probing tasks: with the replay population's own primitives, 44 of 46 predictions fall inside them, so the magnitude gap comes from primitive estimation rather than the recurrence.

B Formal Router Model and Comparison

Router policies and actions. A decision round is one decision with a measurable outcome; its observable state x_n includes the task and the available history, and an information state I_n holds records from earlier rounds. A router g selects an action and updates its information from feedback f_n:

$$u_n = g(x_n, I_n), \qquad I_{n+1} = \mathrm{Upd}(I_n, x_n, u_n, f_n). \tag{13}$$

An action is an execution plan: which capabilities run (models, tools, skills, or submodels), how (one path, a parallel portfolio, or a sequence of attempts), and a commit rule that ends the task (Table 17). An ex-ante router commits before execution, possibly after paying for a routing signal. A verify-then-escalate router runs and checks a cheap primary, then escalates on rejection. An agentic router may dispatch one model per task and retain its verified outcome (Zhou et al., 2026), or re-route and retry within the agent loop (Liu et al., 2026). An intra-model router selects submodels, steps, or token budgets, with full computation as a possible reference (Liang et al., 2026; Fu et al., 2026a). All four consume resources and deliver value; the cascade is the running example because its costs are explicit. Theorem 2 requires a policy-independent state distribution. Within-task retries after failure are instead covered by the escalation recursion of Proposition 10.

Comparing routing policies. On a common task population, let (V, D) be the value delivered and resource consumed under the router and (Y_B, C_B) those of a reference policy, all nonnegative and integrable, with $\bar{c} = \mathbb{E}[C_B] > 0$, $\mathbb{E}[D] > 0$, $\bar{v} = \mathbb{E}[Y_B]$, and $\eta_B = \bar{v}/\bar{c}$. For a stationary ergodic sequence of rounds, the long-run value per resource is

$$\eta = \lim_{N \to \infty} \frac{\sum_{n<N} V_n}{\sum_{n<N} D_n} = \frac{\mathbb{E}[V]}{\mathbb{E}[D]} \quad \text{almost surely}, \tag{14}$$

a ratio of aggregates, not the average of V_n/D_n.

Regime	Commits	Evidence	Fallback or reference	Examples
Ex-ante dispatch	before execution	none, or a paid routing signal Z	none	RouteLLM, SCOPE-Router
Verify-then-escalate	after a cheap attempt	explicit verifier, H	a fixed stronger path	FrugalGPT, AutoMix, LLM cascades
In-loop agentic	per task, or at each agent step	implicit and execution-grounded: sandbox runs, tests	none within the task (the record improves later tasks), or any re-route/retry at a step	ACRouter (per task), harness-native routing (per step)
Intra-model	per submodel, step, or token	task loss or reward in training	full computation as reference	VIP-Router, R2R

Table 17: Four routing regimes under one accounting. An executed fallback and a comparison reference play different roles.

Consequences of the reference-relative gain. Lemma 1 is stated in Section 2. Its definitions give $\mathbb{E}[V] = \bar{v} + Q$ and $\mathbb{E}[D] = \bar{c} - \mathcal{S}$, and also

$$Q + \eta_B \mathcal{S} = \mathbb{E}[V - \eta_B D], \qquad \mathbb{E}[V - \lambda D] = \mathbb{E}[D]\,(\eta - \lambda) \quad (\lambda > 0). \tag{15}$$

These identities specialize the common comparison to the routing regimes below; the quality floor remains $Q \geq -\epsilon$.

Scope of the ratio objective. The ratio η measures delivered value per unit of a specified resource. Its use as a decision objective requires that the chosen value zero and resource unit have operational meaning. Here the score zero is the benchmark's zero credit, and scores measure benchmark performance rather than a universal utility. A common offset changes ratio comparisons through η_B; at a fixed exogenous price λ, the policy difference $J_\lambda(g) - J_\lambda(B) = Q + \lambda\mathcal{S}$ is invariant to that offset. The separate floor $Q \geq -\epsilon$ is also invariant. Appendix G.3 gives the shift formula and an example.

The resource D is scalar: dollars, generated tokens, and elapsed seconds define separate comparisons. They are not added in their original units. Multiple resources $R_1, \ldots, R_d$ can instead enter the explicitly priced objective $\mathbb{E}[V] - \sum_{j=1}^{d} \lambda_j \mathbb{E}[R_j]$, with each λ_j stated in value per unit of its resource. A memory-capacity limit, such as peak VRAM, can restrict the feasible action menu without being converted into a per-task charge. Parallel elapsed time is measured from completion times, not obtained by adding call durations. Every efficiency criterion below specializes (15) against a named reference (Appendix G.2); Theorem 2 additionally imposes the quality floor.

Verified execution with escalation. The cascade rule (choose a primary and a fallback plan, run and verify the primary, commit it on acceptance $H = 1$, otherwise ($H = 0$) run the fallback, then record the outcomes) specializes the router. An output is *sufficient* when the verifier accepts it as adequate for the task; a *hit* is an accepted primary output, and a *miss* invokes the fallback. Let Y_P be the primary value, (Y_F, C_F) the fallback value and *additional* resource if invoked, and A the primary cost including routing, aggregation, verification, and update resources. The verifier may be a model judge or an execution check such as a sandbox run or a test suite; the accounting is the same, its cost enters A, and its false accepts enter Q. Then

$$V = HY_P + (1 - H)Y_F, \qquad D = A + (1 - H)C_F, \tag{16}$$

$$Q = \mathbb{E}[H(Y_P - Y_B) + (1 - H)(Y_F - Y_B)]\,, \qquad \mathcal{S} = \mathbb{E}[HC_B - A - (1 - H)(C_F - C_B)]\,. \tag{17}$$

Here $\mathbb{E}[HC_B]$ is the avoided reference work and A includes gate calls, routing signals, and lost cache discounts (Appendix G.8); no independence between H and the reference outcomes is assumed.

The homogeneous cascade. With hit probability p, hit and miss values v_h, v_{miss}, primary cost $a > 0$, and fallback cost $b > 0$,

$$\eta(p) = \frac{v_{\text{miss}} + p(v_h - v_{\text{miss}})}{a + (1 - p)b}. \tag{18}$$

If the fallback restores the reference on every miss and the reference's conditional means do not depend on H (a reference-restoring fallback), then $v_{\text{miss}} = \bar{v}$, $b = \bar{c}$, and for $v_h, \bar{v} > 0$ the cascade is at least as efficient as the reference if and only if

$$p \geq \frac{\bar{v}}{v_h} \frac{a}{b}. \tag{19}$$

With $v_h = \bar{v}$ and $b = \bar{c}$, a tenfold price advantage needs 10% acceptance. The floor holds only without selection: if accepted tasks are those the reference also solves cheaply, so that $\bar{c}_h = \mathbb{E}[C_B \mid H = 1] < \bar{c}$ and $\bar{v}_h = \mathbb{E}[Y_B \mid H = 1]$, the condition is $p(v_h - \bar{v}_h) + \eta_B(p\bar{c}_h - a) \geq 0$ (Appendix G.6); when the accepted output matches the reference on the same tasks ($v_h = \bar{v}_h$), the floor is $a/\bar{c}_h$, or 40% at $\bar{c}_h = \bar{c}/4$. The normalized efficiency

$$\frac{\eta}{\eta_B} = \frac{pG + 1 - p}{1/P + 1 - p}, \qquad G = \frac{v_h}{\bar{v}},\ P = \frac{b}{a}, \tag{20}$$

separates quality retention from cost ratio; an ex-ante router, lacking the wasted primary term, is a different specialization (Appendix G.4).

The verifier's operating point. Let $H = \mathbf{1}\{\Sigma \geq \theta\}$, let π_0 be the primary's correctness probability, and let $\mathrm{TPR}(\theta), \mathrm{FPR}(\theta)$ be acceptance probabilities conditional on correctness or error. With binary value, $p = \pi_0 \mathrm{TPR} + (1-\pi_0)\mathrm{FPR}$ and $pv_h = \pi_0 \mathrm{TPR}$. If reference success is independent of primary correctness, its cost is constant, and Σ is conditionally independent of Y_B given primary correctness, the homogeneous floor reduces to $\pi_0 \mathrm{TPR}(\theta) b \geq \bar{v} a$; only under these assumptions does the false-accept term cancel. Accepting everything is then efficiency-optimal whenever $a/b \leq \pi_0/\bar{v}$, but can violate the separate quality floor. Appendix G.5 gives the general criterion with the false-accept term and the quality-constrained operating point.

C Formal Results for Model Combinations

Let E_m be the event that candidate m supplies a sufficient output on a shared task, and let an exact selecting verifier pick a sufficient candidate whenever one exists. For a portfolio S,

$$p(S) = \mathbb{P}\Big(\bigcup_{m \in S} E_m\Big), \qquad r_m(S) = \mathbb{P}\Big(E_m \cap \bigcap_{i \in S} E_i^c\Big); \tag{21}$$

the residual coverage $r_m(S)$ is the extra success probability supplied by m, under arbitrary correlations.

Proposition 5 (Exact paired portfolio comparison)**.** *Let (V_S, D_S) and (V_{S^+}, D_{S^+}) be the delivered values and complete costs of two portfolio policies on the same task population, where $S^+ = S \cup \{m\}$ and both expected costs are positive. Define*

$$\begin{aligned} \Delta\mathcal{V}_m(S) &= \mathbb{E}[V_{S^+} - V_S], \qquad \Delta\mathcal{D}_m(S) = \mathbb{E}[D_{S^+} - D_S], \\ G_m(S) &= \Delta\mathcal{V}_m(S) - \eta(S)\Delta\mathcal{D}_m(S). \end{aligned} \tag{22}$$

Then

$$\eta(S^+) - \eta(S) = \frac{G_m(S)}{\mathbb{E}[D_{S^+}]}, \tag{23}$$

so adding m improves efficiency if and only if $G_m(S) > 0$. Ranking candidates by efficiency improvement uses $G_m(S)/\mathbb{E}[D_{S^+}]$, since their enlarged-portfolio costs may differ. The identity permits changes in accepted quality, fallback task composition, and verifier behavior; it does not require common conditional values. The quality requirement remains separate. (Proof in Appendix L.1.)

Proposition 6 (Efficiency-aware residual coverage)**.** *Suppose the verifier is exact (it selects a sufficient candidate whenever one exists), the compared portfolios have common conditional values v_h, v_{miss} and fallback cost b, and they follow* (18) *with primary cost $a(S)$ and efficiency $\eta(S)$. Let $d_m(S) = a(S \cup \{m\}) - a(S)$, including any change in aggregation and verification cost. Adding m strictly improves efficiency if and only if*

$$r_m(S)\big[v_h - v_{\text{miss}} + \eta(S)b\big] > \eta(S) d_m(S). \tag{24}$$

(Proof in Appendix L.2.)

This is the common-conditional-value specialization of Proposition 5. When these quantities shift, let $p = p(S)$, $r = r_m(S)$, and let $\delta v_h, \delta v_{\text{miss}}, \delta b$ denote their enlarged-minus-current differences. With $N_m(S) = r[v_h - v_{\text{miss}} + \eta(S)b] - \eta(S)d_m(S)$, the exact gain is

$$G_m(S) = N_m(S) + (p + r)\delta v_h + (1 - p - r)\big[\delta v_{\text{miss}} - \eta(S)\delta b\big]. \tag{25}$$

Thus conditional-value shifts change the residual approximation, not the paired comparison. Appendix L.1 derives the correction. Estimating either score on fitting tasks and evaluating the selected additions on held-out tasks tests prediction; evaluating the exact identity on the same observations only checks accounting. Under exact selection, $r_m(S) = p(S^+) - p(S)$ is a joint property that marginal scores cannot identify:

Proposition 7 (A limitation of marginal-quality selection)**.** *Let a mandatory primary m_0 be paired with one of two candidates m_1, m_2, all of cost $c > 0$, with a perfect fallback of cost $b > 0$ and unit value for every completed round. Two task distributions with identical individual success probabilities $(1/2, 1/2, 1/2)$ exist such that the optimal pair costs $2c$ while every possibly randomized selector fed only these marginals has worst-case expected cost at least $2c + b/4$. (Proof in Appendix L.3; Remark 23 gives the identifiable range of $r_m(S)$.)*

Steering and allocation. A role or context view can steer a candidate toward the portfolio's failures, improving conditional coverage if and only if the receiving answer region has greater residual sufficiency (Proposition 24, Appendix L.4); since $p(S)$ is monotone submodular, greedy and knapsack guarantees apply to maximizing coverage itself (Nemhauser et al., 1978; Sviridenko, 2004), though not to $\eta(S)$ or the priced objective J_λ, while adaptive selection needs adaptive submodularity (Golovin & Krause, 2011) (Appendix J.4).

D Formal Results for Routing Decisions and Information

A router also decides whether to attempt its primary, dispatch directly to a stronger path, or pay for information that changes this choice. If the primary costs $a > 0$, is sufficient with probability p, and delivers v_h on a hit, while a miss adds fallback cost b and yields the direct path's value v_b, define for $\lambda > 0$

$$W_\lambda = \lambda b + v_h - v_b, \tag{26}$$

the value of turning a miss into a hit.

Proposition 8 (Attempt-then-escalate threshold)**.** *Attempting the primary first is at least as good as direct fallback if and only if $pW_\lambda \geq \lambda a$. If $W_\lambda > 0$, this is $p \geq \lambda a / W_\lambda$; if $W_\lambda \leq 0$, attempting the primary is strictly worse. A routing refinement costing k that increases p by Δp while preserving the other quantities does not lower the attempt-then-escalate objective if and only if $\Delta p\, W_\lambda \geq \lambda k$; it is worthwhile overall only if the refined cascade also clears the direct-path comparison. (Proof in Appendix H.1.)*

Here b and v_b are miss-conditional means. If the reference's outcomes depend on acceptance, a hit saves the reference's cost on the accepted tasks, $\bar{c}_h$, rather than b; then $W_\lambda = \lambda \bar{c}_h + v_h - \bar{v}_h$ and the band (28) below holds with $\bar{c}_h$ in place of b (Appendix G.6).

Optimal routing and the attempt threshold. Theorem 2, stated in Section 2, characterizes the optimal policy for a fixed finite action menu. For the menu {direct, attempt-then-escalate}, its unconstrained action rule is $p(x)W_{\eta^\star}(x) \geq \eta^\star a(x)$. Under the quality floor, part (iii) gives Proposition 8 at resource price $\eta^\star_\epsilon/(1 + \kappa)$. Verifier operating points can also enter the action menu. The proof and the finite-step optimization argument are in Appendix H.3.

Paying for a more informed routing decision. Suppose accepted outputs and the fallback preserve value, the primary is sufficient with probability s, $0 < a < b$, and exact revelation of the sufficiency event costs $k \geq 0$; the three choices cost

$$C_{\text{cascade}}(s) = a + (1 - s)b, \qquad C_{\text{direct}}(s) = b, \qquad C_{\text{info}}(s) = k + sa + (1 - s)b, \tag{27}$$

the informed choice avoiding an unsuccessful attempt but paying for a successful one.

Proposition 9 (The routing-information band)**.** *Exact routing information is strictly cheaper than both uninformed choices if and only if*

$$\frac{k}{b-a} < s < 1 - \frac{k}{a}. \tag{28}$$

This band is nonempty if and only if $k < a(b-a)/b$, and it then contains the uninformed crossover $s = a/b$. (Proof in Appendix H.5.)

For an imperfect experiment Z with posterior s_Z, the net value is

$$\min\{C_{\text{cascade}}(s), b\} - \mathbb{E}\min\{C_{\text{cascade}}(s_Z), b\} - k; \tag{29}$$

since $\mathbb{E}[s_Z] = s$ and exact revelation dominates any experiment about the same event (Blackwell, 1953), the band bounds where a learned router can pay for itself. Exact information pays only if $k < a$, and at equal cost it beats raising sufficiency by Δp if and only if $b\,\Delta p < a(1-s)$ (Corollary 19; Appendix H.6); using a learned score at the wrong decision threshold can be worse than no signal even at $k = 0$.

Escalation inside a loop. Let a task allow up to K attempts. In a fixed sequence summarized by the preceding failures, a_k is the expected incremental cost of attempt k conditional on reaching it, including verification; p_k is its conditional acceptance probability. Costs may differ across stages and include the longer input context, generated output, tools, and cache effects at that stage. In the shared-value specialization, an accepted attempt delivers v_h, and the terminal path has cost b and value v_b. Write $J^{\text{direct}} = v_b - \lambda b$ and let Π_k be the optimal remaining J_λ after $k-1$ failures, with $\Pi_{K+1} = J^{\text{direct}}$.

Proposition 10 (Escalation decisions inside a loop)**.** *The remaining return is*

$$\Pi_k = \max\{J^{\text{direct}},\, p_k v_h - \lambda a_k + (1-p_k)\Pi_{k+1}\}. \tag{30}$$

Attempt k is optimal if and only if $p_k W_\lambda + (1-p_k)\Delta_{k+1} \geq \lambda a_k$, where $\Delta_{k+1} = \Pi_{k+1} - J^{\text{direct}} \geq 0$. For the last attempt $\Delta_{K+1} = 0$ and the rule is Proposition 8; the option value Δ_{k+1} of later attempts can only lower the bar. ACRouter's per-task reward $\epsilon_1 s_i + \epsilon_2 \kappa_i$ for verified score s_i and cost κ_i (Zhou et al., 2026), with $\epsilon_1 > 0 > \epsilon_2$, is $\epsilon_1 J_\lambda$ with $\lambda = -\epsilon_2/\epsilon_1$, with $J_\lambda(x,u)$ the objective of action u in state x, and cumulative regret against a per-task oracle is $\epsilon_1 \sum_i [\max_u J_\lambda(x_i,u) - J_\lambda(x_i,u_i)]$. (Proof in Appendix H.2.)

When observed feedback distinguishes histories with the same number of failures, the state must retain that history. For history $\mathfrak{h}$ and next action u, let $a(\mathfrak{h},u)$ be the conditional expected incremental cost, $p(\mathfrak{h},u)$ the acceptance probability, and $v_{\text{acc}}(\mathfrak{h},u)$ the accepted value. The finite-horizon recursion is

$$\begin{aligned} J^{\text{attempt}}(\mathfrak{h},u) &= -\lambda a(\mathfrak{h},u) + p(\mathfrak{h},u) v_{\text{acc}}(\mathfrak{h},u) \\ &\quad + (1-p(\mathfrak{h},u))\mathbb{E}[\Pi(\mathfrak{h}') \mid \mathfrak{h}, u, \text{rejection}], \\ \Pi(\mathfrak{h}) &= \max\Big\{ J^{\text{direct}}(\mathfrak{h}),\ \max_{u \in \mathcal{U}(\mathfrak{h})} J^{\text{attempt}}(\mathfrak{h},u) \Big\}. \end{aligned} \tag{31}$$

Here $J^{\text{direct}}(\mathfrak{h})$ also uses the fallback's history-conditional value and cost. Previously incurred costs are sunk at this decision; every subsequent call is charged at its actual context and execution state. Equation (31) reduces to Proposition 10 when the failure index is sufficient and the shared-value assumptions hold (Appendix H.2). A step-level agentic router (Liu et al., 2026) can use this history state, whereas ACRouter's released in-distribution mode (Zhou et al., 2026) dispatches once without a terminal fallback; its verification writes a record for later tasks.

Parallelism changes the resource being optimized. Under concurrent execution of primary and fallback, latency is set by completion times rather than summed charges, and cancellation saves only the unbilled suffix; Appendix I.2 gives the latency and dollar criteria.

E What Should a Router Remember?

A stateful router uses past outcomes through I in (13). ACRouter stores the chosen model, verified score, cost, and execution trace in a bounded embedding-keyed store and retrieves nearest neighbors (Zhou et al., 2026). We study an idealized partition into fixed cells that share a record, with retention as in Section E.1.

A record process. Tasks fall into J cells arriving independently with probabilities $\nu_j > 0$. At the start of a round, each cell is uninformed (U) or holds an informative (I) or erroneous (W) record. After the round's decision, each record expires independently with probability δ_j, and the arriving cell receives an informative write with probability ω_j or an erroneous write with probability ψ_j, replacing its old record. Write types are independent of the past and of the current record (Remarks 25 and 26 and Appendix M.4 relax this). Set

$$r_j = \omega_j + \psi_j > 0, \qquad \alpha_j = (1-\delta_j)(1-\nu_j r_j) < 1. \tag{32}$$

A record written after round n first affects round $n+1$; v_j^s, c_j^s denote the router's expected value and cost in cell j and record state $s \in \{U, I, W\}$, including the record update.

Measuring a record by its gain. The uninformed reference router, which collects records at the same cost but always routes as if uninformed, has efficiency

$$\eta_U = \frac{\sum_j \nu_j v_j^U}{\sum_j \nu_j c_j^U}. \tag{33}$$

Assume its denominator is positive, let $\bar{c}_\infty > 0$ and η_∞ be the stationary expected cost and efficiency, and define the record gain and the stationary gain of a cell by

$$g_j^s = (v_j^s - v_j^U) - \eta_U(c_j^s - c_j^U), \quad s \in \{I, W\}, \tag{34}$$

$$G_j = \frac{\nu_j(\omega_j g_j^I + \psi_j g_j^W)}{1-\alpha_j}. \tag{35}$$

Proposition 11 (Transient and stationary memory gains)**.** *From an empty store, the probabilities of informative and erroneous records before round $n \geq 0$ are*

$$i_{j,n} = \frac{\nu_j \omega_j}{1-\alpha_j}(1-\alpha_j^n), \qquad w_{j,n} = \frac{\nu_j \psi_j}{1-\alpha_j}(1-\alpha_j^n). \tag{36}$$

The stationary efficiency satisfies $\eta_\infty \geq \eta_U$ if and only if $\sum_j \nu_j G_j \geq 0$. At every horizon N,

$$\sum_{n=0}^{N-1} \mathbb{E}[V_n - \eta_U D_n] = \sum_j \nu_j G_j \left[N - \frac{1-\alpha_j^N}{1-\alpha_j} \right]. \tag{37}$$

If $g_j^I > 0 > g_j^W$, cell j contributes a positive stationary gain if and only if

$$\frac{\psi_j}{r_j} < \frac{g_j^I}{g_j^I - g_j^W}. \tag{38}$$

(Proof in Appendix M.1.)

The recurrence $i_{j,n+1} = \nu_j \omega_j + \alpha_j i_{j,n}$ (a good record is freshly written or survives without expiration or replacement) and the summed gains give the proposition. Under $g_j^I > 0 > g_j^W$, write quality fixes the sign of a cell's contribution and write frequency and expiration fix its weight and rate α_j, so heterogeneous weights can flip the sign of the total and improvement need not be monotone in time; the statement concerns expected aggregates, not a Markov hit sequence.

Proposition 12 (Records inside a verified cascade)**.** *Suppose that in cell j the router runs the cascade of Section 2 with record-independent costs a_j, b_j and values v_h, v_b, and that a record changes only the hit probability, p_j^s for $s \in \{U, I, W\}$. Then $v_j^s = p_j^s v_h + (1-p_j^s)v_b$, $c_j^s = a_j + (1-p_j^s)b_j$, and $g_j^s = (p_j^s - p_j^U)W_{\eta_U,j}$ with $W_{\eta_U,j} = \eta_U b_j + v_h - v_b$. If $W_{\eta_U,j} > 0$, condition* (38) *becomes $\psi_j/r_j < \Delta_j^I/(\Delta_j^I - \Delta_j^W)$ with $\Delta_j^s = p_j^s - p_j^U$. If instead a record is served without verification at lookup cost c_ℓ, an erroneous record (of value zero) has $g_j^W = \eta_U(c_j^U - c_\ell) - v_j^U$, which is nonnegative whenever the cell's uninformed efficiency v_j^U/c_j^U is at most $\eta_U(1 - c_\ell/c_j^U)$; the hypothesis $g_j^W < 0$ of Proposition 11 then fails. (Proof in Appendix M.2.)*

When a record is written whenever the primary is accepted and is erroneous if and only if the accepted output is worse than the reference, $\psi_j/r_j = \mathbb{P}(Y_P < Y_B \mid H = 1, \text{cell } j)$. An established lower

bound τ on the conditional probability of $Y_P \geq Y_B$ for accepted writes gives $\psi_j/r_j \leq 1-\tau$ (Appendix I.1). This is a reliability bound, distinct from the numerical verifier-score cutoff θ. Against a reference without a store, with $J_U(\lambda) = \sum_j \nu_j(v_j^U - \lambda c_j^U)$ and (34) evaluated at that reference's efficiency λ, the stationary condition becomes

$$J_U(\lambda) + \sum_j \nu_j G_j(\lambda) \geq 0; \tag{39}$$

if the uninformed-state process matches a no-memory reference of efficiency η_0 except for a maintenance cost h per round, then $J_U(\eta_0) = -\eta_0 h$, so the records must deliver at least $\eta_0 h$ in additional gain (Appendix M.5).

E.1 Capacity and the Tail of Task Frequencies

A record expiring after E global rounds (a time-to-live, TTL) gives cell j stationary coverage (40); an event FIFO of the last M write events, duplicates included, with $\beta = \sum_j \nu_j r_j$ and the newest matching record consulted, gives (41):

$$\chi_j^{\mathrm{TTL}} = 1 - (1-\nu_j r_j)^E, \tag{40}$$

$$\chi_j^{\mathrm{FIFO}} = 1 - (1-\nu_j r_j/\beta)^M. \tag{41}$$

Both have informative fraction ω_j/r_j given coverage, and substituting the mean span M/β for E in (40) underestimates FIFO coverage (Appendix M.6). For clean writes ($\psi_j = 0$) with a common write probability ω and a common informed-minus-uninformed hit gain Δ, the uncovered hit deficit at TTL horizon E is $L_E = \Delta \sum_j \nu_j(1-\omega\nu_j)^E$; from an empty store without expiration the deficit in round n is the same L_n, with cumulative version $R_N = \sum_{n<N} L_n$.

Proposition 13 (Heavy-tailed information deficit)**.** *Suppose cells are countably infinite with $\nu_j = j^{-z}/\zeta(z)$, $z > 1$. Writes are clean with constant probability $\omega \in (0,1]$, and every cell has the same positive informed-minus-uninformed hit gain Δ. Under a TTL of E rounds, the stationary uncovered hit deficit is $L_E = \Theta(E^{-(z-1)/z})$. From an empty store without expiration, the cumulative uncovered hit deficit over N rounds is $R_N = \Theta(N^{1/z})$. (Proof in Appendix M.7.)*

Multiplied by W_{η_U}, L_E and R_N are the shortfalls of (37) under Proposition 12. The exponents and constants are the occupancy asymptotics of Karlin (1967) and Gnedin et al. (2007) applied to write events, and the TTL formula is the independent-reference miss ratio of Fagin (1977); the routing content is the contrast with finitely many cells, whose cumulative loss stays bounded (Corollary 27 restates the law in stored records; Remark 28 gives break-even horizons).

E.2 Allocating Router Memory Across Task Families

For independent task families with separate stores and increasing, strictly concave coverage curves $p_k(F_k)$, weights w_k, and unit prices γ_k, a budget is optimally allocated by equalizing weighted marginal coverage per unit price, $w_k p_k'(F_k)/\gamma_k = \Lambda$ on funded families ($F_k > 0$; Appendix J). For J_λ the weight of a family routed cheap-first is $w_k = \nu_k W_{\lambda,k}$; because its benefit $\nu_k \max\{0, p_k(F_k)W_{\lambda,k} - \lambda a_k\}$ is not concave in F_k, the budget problem (56) of Appendix J is the inner problem for a fixed set of families routed cheap-first, whose outer choice follows Theorem 2 (Appendix J.3). The exponent depends on what capacity counts: for a retention horizon or an event FIFO length, Proposition 13 gives $1 - p_k \approx d_k(1+F_k)^{-(1-1/z)}$ for a family constant d_k and capacity proportional to $(w_k d_k/\gamma_k)^{z/(2z-1)}$ for large funded allocations; for stored records the deficit is $\Theta(F_k^{-(z-1)})$ (Corollary 27) and capacity scales as $(w_k d_k/\gamma_k)^{1/z}$.

F Notation

Tables 18 and 19 group symbols by model; auxiliary symbols are defined where they are used; the final column specifies each definition's scope. Probability is $\mathbb{P}$, expectation is $\mathbb{E}$, the verifier score is Σ, the portfolio is S, and savings are $\mathcal{S}$. Italic and upright U, I, W denote the same record states.

Table 18: Symbols for the router model, model combinations, and routing decisions.

Symbol	Meaning	Where
Router outcomes and verification		
x_n, I_n, u_n, f_n	observable state, information state (records), action and feedback of round n	Eq. (13)
g	router policy $u_n = g(x_n, I_n)$	Sec. 2
$V,\ D$	value delivered and resource consumed by the router in a round	Sec. 2
Y_B, C_B; $\bar{v}$, $\bar{c}$; η_B	reference value and cost, their means, reference efficiency $\bar{v}/\bar{c}$	Sec. 2
η	efficiency $\mathbb{E}[V]/\mathbb{E}[D]$, long-run value per resource	Eq. (14)
$Q,\ \mathcal{S}$	quality change $\mathbb{E}[V - Y_B]$ and resource saving $\mathbb{E}[C_B - D]$; $Q + \eta_B \mathcal{S}$ is the net gain	Lem. 1
ϵ	tolerance of the quality floor $\mathbb{E}[V] \geq \bar{v} - \epsilon$	Lem. 1
Y_P, H, Y_F, C_F, A	primary value, acceptance (hit) indicator, fallback value and additional resource, primary and routing cost	Eq. (16)
p, v_h, v_{miss}, a, b	hit probability, hit- and miss-conditional values, primary cost, miss-conditional fallback cost	Eq. (18)
$G,\ P$	$v_h/\bar{v}$ and b/a	Eq. (20)
$\bar{v}_h$, $\bar{v}_{\text{miss}}$, $\bar{c}_h$, $\bar{c}_{\text{miss}}$	reference value and cost means conditional on $H = 1$ and $H = 0$	App. G.6
e	erroneous-acceptance probability $\mathbb{P}(H = 1, Y_P < Y_B)$	App. G.1
$\Sigma,\ \theta$	verifier score and acceptance threshold, $H = \mathbf{1}\{\Sigma \geq \theta\}$	Sec. 2
π_0, $\text{TPR}(\theta)$, $\text{FPR}(\theta)$	probability that the primary is correct; acceptance probabilities given a correct and an incorrect primary	App. B, App. G.5
β_1, β_0	reference success probabilities given a correct and an incorrect primary	App. G.5
Portfolios and complementarity		
E_m, S, $p(S)$, $r_m(S)$	success event of candidate m, portfolio, its coverage, residual coverage of m given S	Eq. (21)
$a(S)$, $d_m(S)$, $\eta(S)$	primary cost of portfolio S, added cost of candidate m, efficiency of S	Prop. 6
$\Delta\mathcal{V}_m(S)$, $\Delta\mathcal{D}_m(S)$, $G_m(S)$	paired changes in expected value and total cost; exact portfolio gain	Prop. 5
c	cost of each candidate in the marginal-quality counterexample	Prop. 7
ρ, T_0, T_1	steering parameter and the de-emphasized and retained answer-type regions	App. L.4
Continuation, information, and policy optimization		
λ, J_λ, $J_\lambda(x, u)$, $\eta(g)$	resource price; linear objective $\mathbb{E}[V - \lambda D]$, the same objective for action u in state x; efficiency of router g	Sec. 2
W_λ, v_b	margin $\lambda b + v_h - v_b$ (value of turning a miss into a hit); value of the direct path	Eq. (26)
k	cost of a routing refinement or routing signal	Prop. 8
Δp	increase in hit probability bought by a refinement	Prop. 8
s	probability of primary sufficiency	Eq. (27)
C_{cascade}, C_{direct}, C_{info}	expected costs of the two uninformed choices and of the informed choice	Eq. (27)
Z, s_Z	experiment (routing signal) and the posterior sufficiency probability it induces	Eq. (29)
$g(u \mid x)$	randomized action choice in state x	Thm. 2
$\pi(x)$, $\mathcal{U}(x)$, $v(x, u)$, $c(x, u)$	state probabilities, action menu, expected value and cost of action u in state x	Thm. 2
$\Phi(\lambda)$, $\eta^\star$, $\eta^\star_\epsilon$, κ	Dinkelbach function, optimal efficiency, floor-constrained optimum, floor multiplier	Thm. 2
K, a_k, p_k, Π_k, Δ_{k+1}, J^{direct}	attempt count; cost, conditional acceptance, continuation and option values at attempt k; terminal objective	Prop. 10
ϵ_1, ϵ_2; s_i, κ_i	per-task reward weights, verifier score and cost of task i in Zhou et al. (2026)	Prop. 10

Table 19: Symbols of Section E and its replay experiments.

Symbol	Meaning	Where
J, ν_j	number of cells and arrival probability of cell j	Sec. E
$s \in \{U, I, W\}$	record states: uninformed, informative, erroneous	Sec. E
ε	exploration probability of the semi-synthetic logger	Sec. 5
δ_j, ω_j, ψ_j, r_j, α_j	expiration probability, informative and erroneous write probabilities, write rate $\omega_j + \psi_j$, retention factor	Eq. (32)
e_j	erroneous-write fraction ψ_j / r_j of cell j	Sec. 5
v_j^s, c_j^s	conditional value and cost in cell j and record state s	Sec. E
η_U, g_j^s, G_j	efficiency of the uninformed reference, record gain, stationary gain of cell j	Eqs. (33)-(35)
$i_{j,n}$, $w_{j,n}$	probabilities of an informative and of an erroneous record before round n	Eq. (36)
η_∞, $\bar{c}_\infty$	stationary efficiency and stationary expected cost	Prop. 11
$J_U(\lambda)$, h	objective of the uninformed state at price λ; maintenance cost per round	Eq. (39)
η_0, c_0	efficiency and mean cost of a no-memory reference	Eq. (39)
p_j^s, Δ_j^s, $W_{\eta_U, j}$, c_ℓ	hit probability in record state s, its change $p_j^s - p_j^U$, margin at price η_U, lookup cost of an unverified record	Prop. 12
Retention and capacity allocation		
E, M, β	TTL horizon in rounds, FIFO length in write events, total write probability $\sum_j \nu_j r_j$	Eqs. (40), (41)
χ_j	stationary coverage of cell j	Eqs. (40), (41)
z, $\zeta(z)$, κ_z	Zipf exponent, Riemann zeta function, normalizer $1/\zeta(z)$	Prop. 13
ω, Δ	common clean-write probability and informed-minus-uninformed hit gap	Prop. 13
L_E, R_N, $m(E)$	per-round uncovered deficit, cumulative deficit, expected number of occupied cells	Sec. E.1, Cor. 27
F_k, $p_k(F_k)$, γ_k, w_k, d_k	allocation to family k, its coverage curve, unit price, weight, deficit scale	Sec. E.2
Λ, B	budget multiplier and budget	Eq. (56)
r	coverage exponent in $(1+F)^{-r}$	Sec. E.2
K, a_k	number of task families and primary cost of family k	App. J.3

G Router Comparison and Homogeneous Cascades

G.1 Proof of Lemma 1

Proof. For any router, the definitions $Q = \mathbb{E}[V - Y_B]$ and $\mathcal{S} = \mathbb{E}[C_B - D]$ give $\mathbb{E}[V] = \bar{v} + Q$ and $\mathbb{E}[D] = \bar{c} - \mathcal{S}$. Under verified execution with escalation, Equation (16) further gives the pointwise expansions

$$V - Y_B = H(Y_P - Y_B) + (1 - H)(Y_F - Y_B),$$

and

$$C_B - D = HC_B - A - (1 - H)(C_F - C_B).$$

Taking expectations yields Equation (17). In either case, $\bar{c} - \mathcal{S} > 0$ and

$$\eta - \eta_B = \frac{\bar{v} + Q}{\bar{c} - \mathcal{S}} - \frac{\bar{v}}{\bar{c}} = \frac{Q + (\bar{v}/\bar{c})\mathcal{S}}{\bar{c} - \mathcal{S}},$$

which is (4). The denominator is positive, giving the efficiency comparison. For (15), substitute $\bar{v} = \eta_B \bar{c}$:

$$Q + \eta_B \mathcal{S} = (\mathbb{E}[V] - \bar{v}) + \eta_B(\bar{c} - \mathbb{E}[D]) = \mathbb{E}[V - \eta_B D],$$

and, for any λ, $\mathbb{E}[V - \lambda D] = \mathbb{E}[D]\,(\mathbb{E}[V]/\mathbb{E}[D] - \lambda) = \mathbb{E}[D](\eta - \lambda)$. The quality condition follows separately from $\mathbb{E}[V] = \bar{v} + Q$. These equalities require neither independence nor stationarity. For the long-run statement, the ergodic theorem applied to integrable V_n and D_n gives convergence of their sample means to their expectations. Their ratio converges because the limiting mean cost is positive. □

The comparison uses a reference policy on the same task population. If routing changes subsequent within-task states, taking the entire task as the round includes all consequences in (V, D), including both the primary and fallback paths. A null action can enter the action menu; the within-task comparison between attempting and stopping for a fallback is specified in (31). Evaluation on intermediate states visited by the router instead gives a comparison on that particular state distribution.

For example, if fallback restores the reference on every miss, values lie in $[0, 1]$, and $e = \mathbb{P}(H = 1, Y_P < Y_B)$, then $Q \geq -e$. Savings satisfying $\eta_B \mathcal{S} \geq e$ suffice for an efficiency guarantee, while $e \leq \epsilon$ suffices for the quality constraint. Neither conclusion requires the verifier to preserve the reference distribution. Appendix G.5 expresses e through the verifier's false-accept rate.

G.2 How the results specialize the gain identity

Table 20 applies the gain identity against each result's reference and price. The memory row is per cell and record; the aggregate criterion is $\sum_j \nu_j G_j \geq 0$. The latency row measures resource in time, and the information row preserves value. In every case the quality floor $Q \geq -\epsilon$ remains a separate condition.

Table 20: Specializations of the gain identity (15): reference policy and price, the terms Q and $\mathcal{S}$, and the criterion $Q + \lambda \mathcal{S} \geq 0$.

Result	Reference (price λ)	Q	$\mathcal{S}$	Criterion
Eq. (19)	direct path with $\bar{v}, \bar{c} = b$ (η_B)	$p(v_h - \bar{v})$	$pb - a$	$pv_h b \geq \bar{v} a$
Prop. 5	current portfolio S ($\eta(S)$)	$\Delta \mathcal{V}_m(S)$	$-\Delta \mathcal{D}_m(S)$	$G_m(S) \geq 0$
Prop. 6	current portfolio S ($\eta(S)$)	$r_m(S)(v_h - v_{\text{miss}})$	$br_m(S) - d_m(S)$	Eq. (24)
Prop. 8, refinement clause	unrefined cascade (λ)	$\Delta p\,(v_h - v_b)$	$\Delta p\, b - k$	$\Delta p\, W_\lambda \geq \lambda k$
Eq. (34)	uninformed store, state U (η_U)	$v_j^s - v_j^U$	$c_j^U - c_j^s$	$g_j^s > 0$; stationary $\sum_j \nu_j G_j \geq 0$
Eq. (53)	reference path, latency ℓ_b ($\bar{v}/\ell_b$)	$p(v_h - \bar{v})$	$p\ell_b - \ell_\kappa - p\ell_p$	$p(\ell_b v_h - \bar{v}\ell_p) \geq \bar{v}\ell_\kappa$
Prop. 9	cheap-first cascade (any price)	0	$a(1-s) - k$	$s \leq 1 - k/a$

The main clause of Proposition 8 is the first row at an exogenous price: with the direct path as reference, $Q = p(v_h - v_b)$, $\mathcal{S} = pb - a$ and $Q + \lambda \mathcal{S} = pW_\lambda - \lambda a$. Against the direct path, Proposition 9 has $\mathcal{S} = s(b - a) - k$, and the band (28) is the intersection of the two savings conditions.

G.3 Why the quality floor and the value zero matter

The efficiency equivalences of the paper leave the floor $Q \geq -\epsilon$ as a separate condition, and values must be measured on a scale whose zero is meaningful. Two examples distinguish these choices.

Blind acceptance passes the ratio test. Let $a = 1/10$ and $b = \bar{c} = \bar{v} = 1$, and let the primary be correct with probability $q = 3/20$. Accepting every primary output gives $\eta = q/a = 3/2 > \eta_B = 1$, yet $Q = q - 1 = -17/20$. It therefore passes the efficiency comparison while violating every quality floor with $\epsilon < 17/20$. A perfect free verifier instead accepts only correct outputs and restores the reference otherwise, giving $\eta = 20/19$ and $Q = 0$. Theorem 2(iii) enforces this separate quality constraint.

The zero of the value scale. Adding an offset t to every policy's completed-task value leaves Q unchanged but changes the reference ratio to $\eta_B + t/\bar{c}$. Therefore

$$Q + \eta_B^{(t)} \mathcal{S} = Q + \eta_B \mathcal{S} + \frac{t}{\bar{c}} \mathcal{S}, \qquad J_\lambda^{(t)}(g) - J_\lambda^{(t)}(B) = Q + \lambda \mathcal{S}. \tag{42}$$

The first comparison can reverse when costs differ; the second is invariant at a fixed external price λ. The benchmark's original zero-credit score defines the ratio analyses in this paper. In the homogeneous cascade with $a = 1/10$, $b = \bar{c} = 1$ and $p = 1/2$, the values $(v_h, \bar{v}) = (7/10, 1)$ give $Q = -3/20$, $\mathcal{S} = 2/5$ and net gain $1/4 > 0$. Subtracting $7/10$ from every value preserves Q and $\mathcal{S}$ but changes η_B to $3/10$ and the net gain to $-3/100 < 0$. The record threshold (38) has the same dependence through the cost channel of (34); with equal record-state costs that channel vanishes and the threshold is shift-invariant.

G.4 Homogeneous threshold and endpoint bounds

Total expectation gives

$$\eta(p) = \frac{pv_h + (1-p)v_{\text{miss}}}{a + (1-p)b}.$$

Suppose the fallback restores the reference on every miss and the reference's conditional means do not depend on H, so that $v_{\text{miss}} = \bar{v}_{\text{miss}} = \bar{v}$ and $b = \bar{c}_{\text{miss}} = \bar{c}$. This is the reference-restoring fallback of (19); we call it the homogeneous case below. With $v_h, \bar{v}, a, b > 0$,

$$\begin{aligned} \eta(p) \geq \eta_B &\iff b[pv_h + (1-p)\bar{v}] \geq \bar{v}[a + (1-p)b] \\ &\iff pbv_h \geq \bar{v}a. \end{aligned}$$

Strict comparison follows by replacing both inequalities with strict inequalities. A floor above one cannot be attained by an admissible acceptance probability. If the reference means depend on H, the condition is (46) in Appendix G.6.

For $G = v_h/\bar{v}$ and $P = b/a$, write

$$R(p) = \frac{pG + 1 - p}{1/P + 1 - p}, \qquad R'(p) = \frac{G(1+P) - 1}{P(1/P + 1 - p)^2}.$$

Since the derivative has constant sign and the endpoint values are $R(0) = P/(1+P)$ and $R(1) = GP$,

$$\min\left\{GP, \frac{P}{1+P}\right\} \leq R(p) \leq \max\left\{GP, \frac{P}{1+P}\right\}, \quad 0 \leq p \leq 1.$$

Also $R(p) \geq [P/(1+P)](pG + 1 - p)$ because its denominator is at most $(1+P)/P$. When $G(1+P) < 1$, the ratio decreases and remains below one. The crossing $R(p) = 1$ is $p = 1/(GP)$, feasible if and only if $GP \geq 1$.

More generally, holding conditional values and positive resources fixed,

$$\frac{d}{dp}\frac{pv_h + (1-p)v_{\text{miss}}}{pd_h + (1-p)d_m} = \frac{v_h d_m - v_m d_h}{[pd_h + (1-p)d_m]^2}.$$

Thus higher acceptance improves efficiency if and only if a hit has greater value per resource than a miss. Changing the accepted task population need not preserve these conditional means.

For an ex-ante router, let T indicate choosing the primary branch before execution and $t = \mathbb{P}(T)$. With no routing overhead, primary cost a, reference cost b, and conditional values $v_h, \bar{v}$, normalized efficiency is

$$\frac{tG + 1 - t}{ta/b + 1 - t}.$$

The always-paid wasted-primary term is absent. Branch selection T is not the verified acceptance event H.

G.5 Thresholded verifiers and the operating point

Let the verifier accept when a score clears a threshold, $H = \mathbf{1}\{\Sigma \geq \theta\}$, let $Y \in \{0, 1\}$ indicate that the primary output is correct, with $\pi_0 = \mathbb{P}(Y = 1)$, and write $\text{TPR}(\theta) = \mathbb{P}(\Sigma \geq \theta \mid Y = 1)$ and $\text{FPR}(\theta) = \mathbb{P}(\Sigma \geq \theta \mid Y = 0)$; both are nonincreasing in θ. Give a correct output value one and an incorrect output value zero, and let the fallback restore the reference on a miss. Then

$$p(\theta) = \pi_0 \text{TPR}(\theta) + (1 - \pi_0)\text{FPR}(\theta), \qquad p(\theta)\, v_h(\theta) = \mathbb{E}[HY_P] = \pi_0 \text{TPR}(\theta),$$

so the product pv_h that enters the floor (19) counts true accepts only. Moving θ changes p, v_h and the erroneous-acceptance probability e together; the refinement clause of Proposition 8 holds the other quantities fixed and therefore does not describe a change of threshold, which is instead one more action in the menu $\mathcal{U}(x)$ of Theorem 2. Choosing θ is the error-reject trade-off of Chow (1970), priced here by the cascade.

Efficiency. Let $\beta_1 = \mathbb{P}(Y_B = 1 \mid Y = 1)$ and $\beta_0 = \mathbb{P}(Y_B = 1 \mid Y = 0)$ be the reference's success probabilities given a correct and an incorrect primary, so that $\bar{v} = \pi_0\beta_1 + (1-\pi_0)\beta_0$; suppose the reference cost is a constant b and the score Σ is conditionally independent of Y_B given Y. Then $\mathbb{P}(H = 1, Y_B = 1) = \pi_0\beta_1\mathrm{TPR} + (1-\pi_0)\beta_0\mathrm{FPR}$, and with $Q = \mathbb{E}[H(Y_P - Y_B)]$ and $\mathcal{S} = \mathbb{E}[HC_B] - \mathbb{E}[A] = pb - a$,

$$Q = \pi_0(1-\beta_1)\mathrm{TPR} - (1-\pi_0)\beta_0\mathrm{FPR}, \qquad e = (1-\pi_0)\beta_0\mathrm{FPR}, \tag{43}$$

$$Q + \eta_B\mathcal{S} = \pi_0(1-\beta_1+\bar{v})\mathrm{TPR} + (1-\pi_0)(\bar{v}-\beta_0)\mathrm{FPR} - \frac{\bar{v}a}{b}. \tag{44}$$

Efficiency improves if and only if the right side of (44) is positive. In the homogeneous case, $\beta_1 = \beta_0 = \bar{v}$, reference success is independent of the primary's correctness. The FPR term then vanishes and the condition is $\pi_0\mathrm{TPR}(\theta)\, b \geq \bar{v}a$, which is (19) with $pv_h = \pi_0\mathrm{TPR}$. False accepts do not enter the efficiency comparison: an accepted incorrect output costs a instead of $a + b$, and the value it forgoes is value that, at $\beta_0 = \bar{v}$, the reference delivers with the same probability as on any other task. When $\beta_1 > \beta_0$ and $\mathrm{TPR}(\theta) > \mathrm{FPR}(\theta)$, acceptance selects tasks the reference also solves, $\bar{v}_h = \mathbb{P}(Y_B = 1 \mid H = 1) > \bar{v}$, and condition (46) is stricter than (19): it reads $\pi_0\mathrm{TPR}(\theta)\, b - pb(\bar{v}_h - \bar{v}) \geq \bar{v}a$, with the selection term $pb(\bar{v}_h - \bar{v})$ on the left.

Accepting everything. In the homogeneous case, $\eta(\theta) = [\pi_0\mathrm{TPR} + (1-p)\bar{v}]/[a + (1-p)b]$ with $p = p(\theta)$, and accepting every output gives $\eta = \pi_0/a$. Cross-multiplying,

$$\eta(\theta) \leq \frac{\pi_0}{a} \iff a(1-p)\bar{v} \leq \pi_0 a(1-\mathrm{TPR}) + \pi_0(1-p)b.$$

Thus $a\bar{v} \leq \pi_0 b$ suffices for accepting every output to maximize efficiency over all thresholds. More generally, $\pi_0\mathbb{E}[D] - a\mathbb{E}[V]$ is affine on the operating-point square $(\mathrm{TPR}, \mathrm{FPR}) \in [0,1]^2$; checking its four vertices shows that accepting everything dominates this whole square if and only if $a\beta_0 \leq \pi_0 b$. For a fixed score whose operating points cover only part of the square, this condition is sufficient but need not be necessary. The quality floor remains separate: by (43), $Q \geq -\epsilon$ is

$$\mathrm{FPR}(\theta) \leq \frac{\epsilon + \pi_0(1-\beta_1)\mathrm{TPR}(\theta)}{(1-\pi_0)\beta_0}, \tag{45}$$

an upper bound on the false-accept rate; because the right side contains $\mathrm{TPR}(\theta)$, the admissible thresholds need not form an up-set in general, but for a score with a monotone likelihood ratio they form an interval $[\theta_{\min}, \infty)$, so the floor bounds the threshold from below. The sufficient condition $e \leq \epsilon$ of Appendix G.1 is (45) without the TPR term; the two coincide when $\beta_1 = 1$.

An instance. Let $\pi_0 = 3/5$, $\beta_1 = 19/20$, $\beta_0 = 4/5$, $a = 1/10$ and $b = 1$, so $\bar{v} = \eta_B = 89/100$. The score has values $1, 2, 3, 4$ with probabilities $(1, 2, 3, 4)/10$ for correct outputs and $(4, 3, 2, 1)/10$ for incorrect ones. Table 21 shows that accepting everything maximizes efficiency but loses $29/100$ in quality. At $\epsilon = 1/20$, the smallest admissible threshold is $\theta = 4$. Allowing randomization over thresholds and the direct path gives the constrained optimum: weights $6/11$ and $5/11$ on $\theta = 3, 4$ bind the floor at $\mathbb{E}[V] = 21/25$ and yield $\eta = 462/373$. At $\theta = 4$, acceptance selects easier reference tasks ($\bar{v}_h = 13/14 > \bar{v}$), so the homogeneous formula predicts $\eta \approx 1.074$ instead of $87/82 \approx 1.061$.

G.6 State-matched reference values and costs

Suppose fallback restores the reference on misses: $Y_F = Y_B$ and $C_F = C_B$ there. Let $\bar{v}_h, \bar{v}_{\mathrm{miss}}$ and $\bar{c}_h, \bar{c}_{\mathrm{miss}}$ denote the reference value and cost means conditional on $H = 1, 0$, respectively. Let $v_h = \mathbb{E}[Y_P \mid H = 1]$ and $a = \mathbb{E}[A]$. For $0 < p < 1$,

$$Q = p(v_h - \bar{v}_h), \qquad \mathcal{S} = p\bar{c}_h - a,$$

Table 21: Operating points of the instance in Appendix G.5: $\pi_0 = 3/5$, $\beta_1 = 19/20$, $\beta_0 = 4/5$, $a = 1/10$, $b = 1$, $\eta_B = 89/100$.

θ	TPR	FPR	p	Q	e	η	$Q + \eta_B \mathcal{S}$
1	1	1	1	$-29/100$	$8/25$	6	$511/1000$
2	$9/10$	$3/5$	$39/50$	$-33/200$	$24/125$	$145/64$	$2201/5000$
3	$7/10$	$3/10$	$27/50$	$-3/40$	$12/125$	$163/112$	$1583/5000$
4	$2/5$	$1/10$	$7/25$	$-1/50$	$4/125$	$87/82$	$701/5000$
5	0	0	0	0	0	$89/110$	$-89/1000$

while $\bar{v} = p\bar{v}_h + (1-p)\bar{v}_{\text{miss}}$ and $\bar{c} = p\bar{c}_h + (1-p)\bar{c}_{\text{miss}}$. The efficiency condition is

$$p(v_h - \bar{v}_h) + \eta_B(p\bar{c}_h - a) \geq 0. \tag{46}$$

In particular, $v_h \geq \bar{v}_h$ and $a \leq p\bar{c}_h$ suffice. If $\eta_B > 0$, either strict inequality yields strict improvement. The event H may be random conditional on the observable state.

When the reference means do not depend on H, so that $\bar{v}_h = \bar{v}$ and $\bar{c}_h = \bar{c}$, condition (46) reduces to (19) with $b = \bar{c}$: $p(v_h - \bar{v}) + \eta_B(p\bar{c} - a) = pv_h - \bar{v}a/\bar{c}$. If accepted outputs preserve value, $v_h = \bar{v}_h$, the condition is $p \geq a/\bar{c}_h$ whatever $\bar{v}_h$ is: the floor is $a/\bar{c}_h$, and it exceeds $a/\bar{c}$ if and only if acceptance selects tasks that the reference solves cheaply, $\bar{c}_h < \bar{c}$. With $\bar{c}_h = \bar{c}/4$, a tenfold price advantage $a = \bar{c}/10$ requires $p \geq 2/5$ rather than $p \geq 1/10$.

Thresholds and the information band with a correlated reference. The same conditional means give the correlated form of Propositions 8 and 9. With the reference as the direct path, its objective is $J_{\text{direct}} = p\bar{v}_h + (1-p)\bar{v}_{\text{miss}} - \lambda[p\bar{c}_h + (1-p)\bar{c}_{\text{miss}}]$, and attempting the primary first gives $J_{\text{cascade}} = pv_h + (1-p)\bar{v}_{\text{miss}} - \lambda[a + (1-p)\bar{c}_{\text{miss}}]$. Subtracting,

$$J_{\text{cascade}} - J_{\text{direct}} = p\,(\lambda\bar{c}_h + v_h - \bar{v}_h) - \lambda a, \qquad \text{so that} \qquad W_\lambda = \lambda\bar{c}_h + v_h - \bar{v}_h.$$

The miss-conditional quantities $b = \bar{c}_{\text{miss}}$ and $v_b = \bar{v}_{\text{miss}}$ cancel: the margin is priced by what the reference would have spent and delivered on the accepted tasks. When the means are H-independent this is (26). For the routing-information band, let U be the sufficiency event with $\mathbb{P}(U = 1) = s$, let the reference incur cost $\bar{c}_h$ on sufficient tasks and $\bar{c}_{\text{miss}}$ on the others, and let values be preserved. Then $C_{\text{direct}} = s\bar{c}_h + (1-s)\bar{c}_{\text{miss}}$, $C_{\text{cascade}} = a + (1-s)\bar{c}_{\text{miss}}$, $C_{\text{info}} = k + sa + (1-s)\bar{c}_{\text{miss}}$, and

$$C_{\text{direct}} - C_{\text{info}} = s(\bar{c}_h - a) - k, \qquad C_{\text{cascade}} - C_{\text{info}} = a(1-s) - k,$$

and $C_{\text{direct}} - C_{\text{cascade}} = s\bar{c}_h - a$, so (28) holds with b replaced by $\bar{c}_h$ (for $\bar{c}_h > a$), the uninformed crossover is $s = a/\bar{c}_h$, and the band is nonempty if and only if $k < a(\bar{c}_h - a)/\bar{c}_h$.

The same reference-restoring fallback gives

$$\frac{\mathbb{E}[HC_B]}{\mathbb{E}[C_B]} = 1 - \frac{\mathbb{E}[D]}{\mathbb{E}[C_B]} + \frac{\mathbb{E}[A]}{\mathbb{E}[C_B]}.$$

Therefore cost savings lower-bound a fraction weighted by reference cost. They lower-bound ordinary acceptance frequency only under an additional condition such as constant C_B.

An auxiliary routing call need not have an invocation event independent of H. Suppose primary plus gate cost $a = c_p + \kappa_g$ is paid every round, while auxiliary routing cost κ_h is paid only on misses. If the fallback backend restores reference value at reference cost b, its total additional resource is $C_F = b + \kappa_h$, and the comparison gives

$$\eta \geq \eta_B \iff p \geq \frac{\bar{v}(c_p + \kappa_g + \kappa_h)}{bv_h + \bar{v}\kappa_h}.$$

This follows by the same cross-multiplication and incorporates the dependence of average overhead on the miss rate.

G.7 Rejected attempts that leave reusable work

The acceptance floor (19) charges a rejected attempt in full: the fallback then pays the whole reference cost b on top of the primary cost a. In a router that continues from the state a rejected attempt leaves

behind, part of that work is reusable. A failed patch may leave a configured environment, a partial edit, or informative test output; the model that continues does not start from nothing. Let $\sigma \in [0,1]$ be the fraction of the reference's work that remains usable after a rejected attempt, and let $\kappa_s \geq 0$ be the cost of switching to the fallback path (a new context, a cold cache, a hand-over). The fallback then restores the reference value $\bar{v}$ at additional cost $(1-\sigma)b + \kappa_s$ instead of b.

Proposition 14 (Salvageable attempts)**.** *Let the reference cost $b = \bar{c}$ and value $\bar{v}$ be independent of acceptance, let the primary cost a include its verification, let an accepted attempt deliver conditional value v_h with probability p, and let a rejected attempt be followed by a fallback that restores $\bar{v}$ at additional cost $(1-\sigma)b + \kappa_s$. The router is at least as efficient as the reference if and only if*

$$p\, b\, v_h + (1-p)\,\sigma b\,\bar{v} \;\geq\; \bar{v}\,[a + (1-p)\kappa_s]. \tag{47}$$

If $bv_h + \kappa_s\bar{v} > \sigma b\bar{v}$, which holds whenever $v_h \geq \bar{v}$ unless $\sigma = 1$, $\kappa_s = 0$, and $v_h = \bar{v}$, this is

$$p \;\geq\; p_\sigma := \frac{\bar{v}\,(a + \kappa_s - \sigma b)}{bv_h - \sigma b\bar{v} + \kappa_s\bar{v}}, \tag{48}$$

which reduces to (19) *at $\sigma = \kappa_s = 0$. If, in addition to $bv_h + \kappa_s\bar{v} > \sigma b\bar{v}$, also $\sigma b \geq a + \kappa_s$, then $p_\sigma \leq 0$: the router is at least as efficient at every acceptance probability, because a rejected attempt leaves behind more reusable reference work than it costs.*

Proof. Substitute $\mathbb{E}[V] = pv_h + (1-p)\bar{v}$ and $\mathbb{E}[D] = a + (1-p)[(1-\sigma)b + \kappa_s]$ into $b\mathbb{E}[V] \geq \bar{v}\mathbb{E}[D]$. Canceling the common miss terms gives (47), or equivalently

$$p(bv_h - \sigma b\bar{v} + \kappa_s\bar{v}) \geq \bar{v}(a + \kappa_s - \sigma b).$$

Dividing by the positive coefficient of p gives (48). Its numerator is nonpositive exactly when $\sigma b \geq a + \kappa_s$, proving the final assertion. □

With $a = 1/10$ and $b = v_h = \bar{v} = 1$ the floor is 10% without salvage and disappears at $\sigma = 3/10$, $\kappa_s = 1/20$ ($p_\sigma = -1/5$). The salvage fraction is a property of the harness, not of the model: a router that re-routes inside a trajectory keeps the environment and the observations of the failed attempt, whereas a cascade that restarts the task from its prompt has $\sigma = 0$, and a per-task router such as ACRouter (Zhou et al., 2026), which commits its single attempt, has no rejected attempt to salvage and lies outside the proposition. Proposition 14 is the one-attempt case; with several attempts the option value of later attempts lowers the bar further (Proposition 10).

A test suite or sandbox check is a verifier with a fixed operating point, covered by Appendix G.5; its execution cost and latency belong in the attempt cost.

G.8 Gating, auxiliary routing, and model switching

For additive charges, write the always-incurred gate cost as κ_g and an auxiliary routing call's cost as κ_h. If J_h indicates invoking that call, expected routing overhead is

$$\mathbb{E}[\kappa_g + J_h\kappa_h].$$

For constant charges this becomes $\kappa_g + \mathbb{P}(J_h = 1)\kappa_h$. It needs no independence between the call and primary acceptance. Put each charge in A if paid before the primary verdict, or in C_F if incurred only on fallback. The miss-only threshold above is the homogeneous instance.

A change of model can also alter input-cache cost. Suppose a fixed prefix has L tokens, fraction γ is reusable, and cached input has price multiplier $\delta \in [0,1]$. Let $c_m, c_{m'}$ be full input prices per token. Staying on model m, whose cache holds the prefix, costs $c_m[(1-\gamma) + \gamma\delta]L$; switching to a destination m' whose cache does not hold it costs $c_{m'}L$. The immediate input-cost difference is

$$c_{m'}L - c_m[(1-\gamma) + \gamma\delta]L.$$

The destination's lost discount relative to its own cached execution is instead $c_{m'}(1-\delta)\gamma L$. These quantities coincide when prices agree. Switching to a sufficiently cheaper model can yield a negative net difference despite losing its cache discount. Apply the gain identity of Lemma 1 to the costs actually incurred on the same task; do not add a cache penalty again if it is already in the billed input cost.

H THRESHOLDS AND THE VALUE OF ROUTING INFORMATION

H.1 PROOF OF PROPOSITION 8

Here b and v_b are the fallback's miss-conditional cost and value, and they are also the cost and value of the direct path: the reference means do not depend on acceptance. Appendix G.6 treats the correlated case, in which $W_\lambda = \lambda\bar{c}_h + v_h - \bar{v}_h$.

Proof. The direct path has objective $J_{\text{direct}} = v_b - \lambda b$. Attempting the primary first has

$$J_{\text{cascade}} = pv_h + (1-p)v_b - \lambda[a + (1-p)b].$$

Subtracting,

$$J_{\text{cascade}} - J_{\text{direct}} = p(v_h - v_b + \lambda b) - \lambda a = pW_\lambda - \lambda a.$$

This proves the comparison and, for $W_\lambda > 0$, the threshold. If $W_\lambda \le 0$, then the difference is strictly negative because $\lambda, a > 0$.

A routing refinement whose stipulated effects are to change p to $p + \Delta p$ and add cost k changes the attempt-then-escalate objective by

$$\Delta p(v_h - v_b + \lambda b) - \lambda k = \Delta p\, W_\lambda - \lambda k.$$

Its sign proves the second assertion. The clause compares the refined cascade with the unrefined one, both attempting first; it is worthwhile overall only if the refined cascade also clears the direct-path comparison, $(p + \Delta p)W_\lambda \ge \lambda(a + k)$. A refinement can satisfy $\Delta p\, W_\lambda \ge \lambda k$ while the cascade remains inferior to dispatching directly, and then the right decision is direct. The refinement changes the success probability with the other quantities held fixed; it does not model a signal allowing contingent choice between fixed actions, nor a change of the verifier's threshold, which moves p, v_h and e together (Appendix G.5). □

H.2 PROOF OF PROPOSITION 10

A task is handled by at most K attempts and, if all of them fail, by a terminal path of cost b and value v_b. Attempt k has expected incremental cost a_k, including its verification, and is accepted with probability p_k conditional on attempts $1, \dots, k-1$ having failed. An accepted attempt delivers v_h and ends the task. A policy decides, after each failure, whether to make the next attempt or to stop and use the terminal path. In this fixed-sequence specialization, the failure index summarizes the information used for that decision. Write $J^{\text{direct}} = v_b - \lambda b$, let Π_k be the supremum of the remaining $J_\lambda = \mathbb{E}[V - \lambda D]$ over policies from the state in which $k-1$ attempts have failed, and set $\Pi_{K+1} = J^{\text{direct}}$, since after K failures only the terminal path remains. The cost a_k is conditional on the state and may increase with context length or change with model prices, tools, and cache reuse; no stage-independent cost is assumed.

Proof. Recursion. Fix $k \le K$ and the state with $k-1$ failures. Stopping yields J^{direct}. Attempting pays a_k; with probability p_k the attempt is accepted and the task ends with value v_h, and with probability $1 - p_k$ it fails and the process reaches the state with k failures. From there the best continuation is worth Π_{k+1}: the probabilities $p_{k+1}, \dots, p_K$ are by hypothesis already conditional on this failure, so the continuation problem is the one that defines Π_{k+1}. Since J_λ is linear in value and cost, and a_k is paid whatever follows, attempting and then continuing optimally is worth $p_k v_h - \lambda a_k + (1-p_k)\Pi_{k+1}$; no continuation is worth more. A randomized decision yields a convex combination of the two numbers and cannot exceed the larger one. Hence

$$\Pi_k = \max\bigl\{J^{\text{direct}},\ p_k v_h - \lambda a_k + (1-p_k)\Pi_{k+1}\bigr\}, \qquad k = K, K-1, \dots, 1,$$

which is backward induction from $\Pi_{K+1} = J^{\text{direct}}$. The finite-horizon dynamic program has an optimal policy that is deterministic and depends only on k: in each state it takes an action attaining the maximum.

The decision rule. Write $\Delta_{k+1} = \Pi_{k+1} - J^{\text{direct}}$ and $J^{\text{direct}} = p_k J^{\text{direct}} + (1-p_k)J^{\text{direct}}$. Subtracting the two branches,

$$\begin{aligned}\bigl[p_k v_h - \lambda a_k + (1-p_k)\Pi_{k+1}\bigr] - J^{\text{direct}} &= p_k\,(v_h - v_b + \lambda b) - \lambda a_k + (1-p_k)\Delta_{k+1} \\ &= p_k W_\lambda - \lambda a_k + (1-p_k)\Delta_{k+1},\end{aligned}$$

with $W_\lambda = \lambda b + v_h - v_b$. Its sign gives the decision rule. Since stopping is always available, $\Delta_{k+1} \geq 0$, and $\Delta_{K+1} = 0$ recovers Proposition 8 at the last attempt. Earlier attempts receive the nonnegative option value $(1 - p_k)\Delta_{k+1}$. For example, with $v_h = v_b = b = \lambda = 1$, $(p_1, a_1) = (1/10, 1/5)$ and $(p_2, a_2) = (9/10, 1/4)$, the first attempt fails its own threshold but is optimal because $1/10 + (9/10)(13/20) > 1/5$. Retry probabilities and costs are conditional on the preceding failures.

History-dependent execution. For a general history $\mathfrak{h}$, the same conditioning argument gives (31): charge the expected next-call cost $a(\mathfrak{h}, u)$ once, collect the conditional accepted value on success, and average the optimal continuation over the rejection histories $\mathfrak{h}'$. The terminal condition is the history-specific direct-path value at the horizon. Backward induction over the remaining attempts proves the recursion. The next-call charge includes processing the retained context as well as generating and verifying the new output; previously paid calls do not enter the remaining-cost comparison again. Consequently, changing context or fallback composition changes the conditional inputs, rather than invalidating the recursion. The simpler margin W_λ requires the shared-value and terminal-path assumptions stated in Proposition 10.

Regret. Zhou et al. (2026) score task i, handled by model a_i, with the reward $r_i = \epsilon_1 s_i + \epsilon_2 \kappa_i$, where s_i is the verifier's score for the output, κ_i its cost, $\epsilon_1 > 0$ and $\epsilon_2 < 0$, and they define the cumulative regret against a per-task oracle as $\sum_i [\max_j R_{ij} - R_{ia_i}]$, where R_{ij} is the reward model j earns on task i. With $\lambda = -\epsilon_2/\epsilon_1 > 0$,

$$r_i = \epsilon_1 \left(s_i - \lambda \kappa_i\right) = \epsilon_1 \, J_\lambda(x_i, a_i),$$

where $J_\lambda(x_i, u) = s_{iu} - \lambda \kappa_{iu}$ is the score of model u on task i net of its priced cost. The reward is thus J_λ up to the positive factor ϵ_1. This factor changes neither the optimal action nor the sign of any gap, and

$$\sum_i \big[\max_j R_{ij} - R_{ia_i}\big] = \epsilon_1 \sum_i \big[\max_u J_\lambda(x_i, u) - J_\lambda(x_i, u_i)\big],$$

the accumulated per-task gap of the proposition, up to the factor ϵ_1. The identity concerns recorded outcomes, as in an outcome matrix. When outcomes are random, the oracle is a hindsight oracle, and since $\mathbb{E} \max_u \geq \max_u \mathbb{E}$ the expected regret is at least the gap computed from expected values. □

In Zhou et al. (2026), $K = 1$ with no within-task fallback: one model's output is committed and verification writes records for later tasks. Step-level agentic routing permits $K > 1$ (Liu et al., 2026); a single verify-then-escalate cascade is $K = 1$ with a terminal fallback.

H.3 Proof of Theorem 2

The proof applies Dinkelbach's fractional-programming characterization (Dinkelbach, 1967) and the Charnes-Cooper transformation (Charnes & Cooper, 1962) to a fixed distribution of observable states. Sparse randomization under a constraint is classical in constrained control (Beutler & Ross, 1985; Altman, 1999); here it follows directly from the finite linear program with one quality constraint. State frequencies are policy-independent, so this theorem does not solve a general controlled Markov process.

Throughout, x ranges over a finite set $\mathcal{X}$ of observable states with router-independent probabilities $\pi(x) > 0$; each state offers a finite menu $\mathcal{U}(x)$ of actions with expected value $v(x, u)$ and expected cost $c(x, u) > 0$; a router g chooses u with probability $g(u \mid x)$. Let $\mathcal{G}$ be the set of routers, a product of simplices and hence a compact polytope, and put $c_{\min} = \min_{x,u} c(x, u) > 0$. For $g \in \mathcal{G}$,

$$\mathbb{E}_g[V] = \sum_x \pi(x) \sum_{u \in \mathcal{U}(x)} g(u \mid x)\, v(x, u), \qquad \mathbb{E}_g[D] = \sum_x \pi(x) \sum_{u \in \mathcal{U}(x)} g(u \mid x)\, c(x, u) \;\geq\; c_{\min},$$

$\eta(g) = \mathbb{E}_g[V]/\mathbb{E}_g[D]$, and for every real λ

$$\mathbb{E}_g[V - \lambda D] = \mathbb{E}_g[D]\,\big(\eta(g) - \lambda\big). \tag{49}$$

A router is deterministic if every $g(\cdot \mid x)$ is a point mass; there are $\prod_x |\mathcal{U}(x)|$ of them. The reference policy is admissible if it is an element $g_B \in \mathcal{G}$; then $\eta(g_B) = \eta_B$ and $\mathbb{E}_{g_B}[V] = \bar{v}$, so the reference satisfies the floor $\mathbb{E}[V] \geq \bar{v} - \epsilon$ for every $\epsilon \geq 0$.

Proof. Two preliminary facts. (F1) State-wise separation: for any real numbers $\phi(x,u)$, the linear functional $g \mapsto \sum_x \pi(x) \sum_u g(u \mid x)\phi(x,u)$ on $\mathcal{G}$ is maximized exactly by the routers that, in every state x, put all their mass on $\arg\max_u \phi(x,u)$, and its maximum is $\sum_x \pi(x) \max_u \phi(x,u)$. Indeed, for each x the inner sum is a convex combination of the numbers $\phi(x,u)$, hence at most $\max_u \phi(x,u)$, with equality if and only if $g(\cdot \mid x)$ is supported on the maximizers; and $\pi(x) > 0$. In particular a deterministic maximizer exists. (F2) Sign of the Dinkelbach function: applying (F1) with $\phi = v - \lambda c$,

$$\Phi(\lambda) = \sum_x \pi(x) \max_{u \in \mathcal{U}(x)} \big[v(x,u) - \lambda c(x,u)\big] = \max_{g \in \mathcal{G}} \mathbb{E}_g[V - \lambda D], \tag{50}$$

so by (49) and $\mathbb{E}_g[D] > 0$, $\Phi(\lambda) \geq 0$ if and only if some router has $\eta(g) \geq \lambda$, and $\Phi(\lambda) > 0$ if and only if some router has $\eta(g) > \lambda$.

Part (i). Each $\phi_x(\lambda) = \max_u [v(x,u) - \lambda c(x,u)]$ is the maximum of finitely many affine functions of λ with slopes $-c(x,u) < 0$. It is therefore convex, piecewise linear and continuous, and strictly decreasing: if $\lambda < \lambda'$ and u' attains $\phi_x(\lambda')$, then $\phi_x(\lambda) \geq v(x,u') - \lambda c(x,u') > v(x,u') - \lambda' c(x,u') = \phi_x(\lambda')$. Moreover $\phi_x(\lambda) \to +\infty$ as $\lambda \to -\infty$, and $\phi_x(\lambda) \leq \max_u v(x,u) - \lambda c_{\min} \to -\infty$ as $\lambda \to +\infty$. The positive combination $\Phi = \sum_x \pi(x)\phi_x$ inherits all these properties and has a unique root $\eta^\star$. By (F2), $\Phi(\eta^\star) = 0$ yields a router with $\eta(g) \geq \eta^\star$ and forbids $\eta(g) > \eta^\star$; hence $\eta^\star = \max_g \eta(g)$, and the maximum is attained. A router g is efficiency-optimal if and only if $\eta(g) = \eta^\star$, if and only if $\mathbb{E}_g[V - \eta^\star D] = 0 = \Phi(\eta^\star)$ by (49), if and only if g maximizes the $\eta^\star$-objective over $\mathcal{G}$, if and only if, by (F1), in every state it uses only actions maximizing $v(x,u) - \eta^\star c(x,u)$. For the two-action menu of Proposition 8, direct has $(v,c) = (v_b, b)$ and attempt-then-escalate has $(pv_h + (1-p)v_b,\ a + (1-p)b)$, all quantities depending on x; the difference of their λ-objectives is $p(v_h - v_b) - \lambda a + \lambda p b = pW_\lambda - \lambda a$ (Appendix H.1). At $\lambda = \eta^\star$ this is the rule $p(x)W_{\eta^\star}(x) \geq \eta^\star a(x)$, either action being allowed at equality.

Part (ii). Set $\lambda_0 = \eta_B$ and, given λ_t, let g_t be a deterministic maximizer of the λ_t-objective, which exists by (F1), and $\lambda_{t+1} = \eta(g_t)$. By (50) and (49), $\Phi(\lambda_t) = \mathbb{E}_{g_t}[D]\,(\lambda_{t+1} - \lambda_t)$. Admissibility gives $\Phi(\lambda_0) \geq \mathbb{E}_{g_B}[V - \eta_B D] = 0$, and for $t \geq 1$, $\Phi(\lambda_t) \geq \mathbb{E}_{g_{t-1}}[V - \lambda_t D] = 0$ because $\lambda_t = \eta(g_{t-1})$. Hence $\lambda_{t+1} \geq \lambda_t$ at every step, with equality if and only if $\Phi(\lambda_t) = 0$, that is $\lambda_t = \eta^\star$, where the iteration stops. As long as it has not stopped, $\lambda_1 < \lambda_2 < \cdots$ are efficiencies of distinct deterministic routers, of which there are finitely many; so the iteration stops after finitely many steps with $\lambda_T = \eta^\star$. A sharper count follows from the piecewise-linear structure of Φ: if $\lambda_s < \lambda_t$ lay in the interior of the same linear piece, g_s and g_t would share that piece's affine function, giving $\lambda_{s+1} = \lambda_{t+1}$, which strict monotonicity forbids. Each piece interior and each breakpoint therefore holds at most one iterate, and since ϕ_x has at most $|\mathcal{U}(x)| - 1$ breakpoints, the iteration takes at most $1 + 2\sum_x (|\mathcal{U}(x)| - 1)$ steps. Any maximizer of the $\eta^\star$-objective is efficiency-optimal by part (i). For the first step, write $Q_g = \mathbb{E}_g[V] - \bar{v}$ and $\mathcal{S}_g = \bar{c} - \mathbb{E}_g[D]$; since $\bar{v} = \eta_B \bar{c}$, $Q_g + \eta_B \mathcal{S}_g = \mathbb{E}_g[V - \eta_B D]$, which is (15), and (50) at $\lambda = \eta_B$ reads $\Phi(\eta_B) = \max_g [Q_g + \eta_B \mathcal{S}_g]$. The first Dinkelbach step thus maximizes the net gain of Lemma 1 over routers, and $\Phi(\eta_B) > 0$ if and only if some router has positive net gain. Without admissibility the iteration still terminates, since $\Phi(\lambda_t) \geq 0$ for all $t \geq 1$; only the monotone start from η_B uses it.

Part (iii): existence and a router that randomizes in at most one state. Let $\mathcal{G}_\epsilon = \{g \in \mathcal{G} : \mathbb{E}_g[V] \geq \bar{v} - \epsilon\}$, a compact polytope containing g_B. Since η is continuous on $\mathcal{G}$ (its denominator is at least $c_{\min}$), a constrained maximizer exists; let $\eta^\star_\epsilon = \max_{\mathcal{G}_\epsilon} \eta$. To obtain the randomization bound, apply the transformation of Charnes & Cooper (1962): for $g \in \mathcal{G}_\epsilon$ put $t = 1/\mathbb{E}_g[D] > 0$, $y(x,u) = t\,\pi(x)\,g(u \mid x)$ and $\sigma = t\,(\mathbb{E}_g[V] - \bar{v} + \epsilon) \geq 0$. Then (y,t,σ) is feasible for the linear program

$$\begin{aligned} \max \quad & \sum_{x,u} y(x,u)\, v(x,u) \\ \text{subject to} \quad & \sum_{x,u} y(x,u)\, c(x,u) = 1, \qquad \sum_{u \in \mathcal{U}(x)} y(x,u) = \pi(x)\, t \quad (x \in \mathcal{X}), \\ & \sum_{x,u} y(x,u)\, v(x,u) - (\bar{v} - \epsilon)\, t - \sigma = 0, \qquad y \geq 0,\ t \geq 0,\ \sigma \geq 0, \end{aligned} \tag{51}$$

with objective value $\eta(g)$. Conversely, a feasible (y,t,σ) has $t > 0$, since $t = 0$ forces $y = 0$ and contradicts the first constraint, and $g(u \mid x) = y(x,u)/(\pi(x)t)$ is a router with $\mathbb{E}_g[D] = 1/t$,

$\mathbb{E}_g[V] = t^{-1}\sum yv = \bar{v} - \epsilon + \sigma/t \geq \bar{v} - \epsilon$ and $\eta(g) = \sum yv$. Thus (51) is feasible, its value is $\eta^\star_\epsilon$, and it is bounded because $\sum y \leq 1/c_{\min}$. A feasible and bounded linear program in standard form has an optimal basic solution, and a basic solution has at most as many positive variables as there are equality constraints, here $|\mathcal{X}| + 2$. Since $t > 0$, at most $|\mathcal{X}| + 1$ of the variables $y(x,u)$ and σ are positive, and each state has at least one positive $y(x,\cdot)$ because $\sum_u y(x,u) = \pi(x)t > 0$. Therefore either every state has exactly one positive $y(x,\cdot)$, a deterministic router, or exactly one state has two positive entries and $\sigma = 0$. The corresponding optimal router randomizes in at most one state, between two actions, and when it does the floor binds. The same count extends to m linear side constraints, each adding one equality row and one slack: an optimal router then randomizes in at most m states.

Part (iii): the multiplier. Define $\Phi_\epsilon(\lambda) = \max_{g\in\mathcal{G}_\epsilon} \mathbb{E}_g[V - \lambda D]$. As a maximum over a compact set of affine functions of λ with slopes $-\mathbb{E}_g[D] \leq -c_{\min}$, it is convex, continuous and strictly decreasing, tends to $\pm\infty$ as $\lambda \to \mp\infty$, and has a unique root. By (49), $\Phi_\epsilon(\lambda) \geq 0$ if and only if some $g \in \mathcal{G}_\epsilon$ has $\eta(g) \geq \lambda$, so the root is $\eta^\star_\epsilon$. Consequently a router is constrained-optimal if and only if it solves

$$\max_{g\in\mathcal{G}} \; f(g) := \mathbb{E}_g[V - \eta^\star_\epsilon D] \quad \text{subject to} \quad h(g) := \mathbb{E}_g[V] - (\bar{v} - \epsilon) \geq 0, \tag{52}$$

whose optimal value is $\Phi_\epsilon(\eta^\star_\epsilon) = 0$. Both f and h are affine in g and $\mathcal{G}$ is a polytope, so (52) is a linear program; it is feasible (g_B) and bounded. By linear-programming strong duality applied to the single constraint $h \geq 0$, there is a multiplier $\kappa \geq 0$ such that

$$\max_{g\in\mathcal{G}} \big[f(g) + \kappa\, h(g)\big] = 0.$$

If g is optimal for (52), then $h(g) \geq 0$ and $f(g) = 0$, so $0 \leq f(g) + \kappa h(g) \leq 0$: hence $\kappa h(g) = 0$ and g maximizes $f + \kappa h$ over $\mathcal{G}$. (Conversely, a feasible maximizer of $f + \kappa h$ with $\kappa h(g) = 0$ has $f(g) = 0$ and is optimal.) Now $f(g) + \kappa h(g) = \mathbb{E}_g[(1+\kappa)V - \eta^\star_\epsilon D] + \kappa(\epsilon - \bar{v})$, so by (F1) every constrained-optimal router uses, in every state, only actions maximizing $(1+\kappa)v(x,u) - \eta^\star_\epsilon c(x,u)$, and $\kappa\,(\mathbb{E}_g[V] - \bar{v} + \epsilon) = 0$: if $\kappa > 0$ the floor binds at every optimal router. The value $\kappa = 0$ is a valid multiplier if and only if some efficiency-optimal router of part (i) satisfies the floor, and then $\eta^\star_\epsilon = \eta^\star$. For the two-action menu the difference of the modified objectives is

$$(1+\kappa)\,p\,(v_h - v_b) - \eta^\star_\epsilon\,(a - pb) = p\big[\eta^\star_\epsilon b + (1+\kappa)(v_h - v_b)\big] - \eta^\star_\epsilon a,$$

and dividing by $1 + \kappa > 0$ gives $pW_{\lambda'} - \lambda' a$ with $\lambda' = \eta^\star_\epsilon/(1+\kappa)$: this is Proposition 8 at price $\eta^\star_\epsilon/(1+\kappa)$. The floor thus enters only through κ, which raises the value of a hit relative to its cost; a state in which the modified rule ties is the one state in which an optimal router may randomize, with the randomization probability set by the floor. The rule identifies the support of optimal routers; κ and $\eta^\star_\epsilon$ themselves are outputs of the linear program (51). □

A two-state example. Let two states be equally likely, with $p = 9/10$, $a = 1/10$, $b = v_b = 1$, and $v_h = 1/2$ in state 1 and $v_h = 1$ in state 2. The reference dispatches directly in both states. The unconstrained optimum attempts in both, giving $\eta^\star = 31/8$ and quality loss $9/40$. Under a tighter floor, state 2 still attempts; attempting in state 1 with probability ϱ gives $\mathbb{E}[V] = 1 - 9\varrho/40$ and $\mathbb{E}[D] = 3/5 - 2\varrho/5$. Their ratio increases in ϱ, so the constrained optimum binds the floor at $\varrho = 40\epsilon/9$:

$$\eta^\star_\epsilon = \frac{45(1-\epsilon)}{27 - 80\epsilon}, \qquad 0 \leq \epsilon \leq \frac{9}{40}.$$

At $\epsilon = 9/80$, this gives $\varrho = 1/2$, $\eta^\star_\epsilon = 71/32$ and multiplier $\kappa = 53/18$: exactly one state randomizes, as part (iii) asserts.

H.4 Proof of Proposition 4

The reference enters a floor-constrained router twice: as the yardstick $\bar{v}$ that defines Q and the floor, and as an input, through every action that calls it. This section states Proposition 4 in full, adding the purchased regime, the effect of refining states, and a reference-price threshold, proves it, and checks it on the menu of Section 3.1.

Setting and notation. We keep the setting and notation of Appendix H.3: finite states with $\pi(x) > 0$, finite menus $\mathcal{U}(x)$, values $v(x,u)$, costs $c(x,u) > 0$, the router set $\mathcal{G}$, $J_\lambda(g) = \mathbb{E}_g[V] - \lambda \mathbb{E}_g[D]$, $\phi_x(\lambda) = \max_{u \in \mathcal{U}(x)} [v(x,u) - \lambda c(x,u)]$, $\Phi = \sum_x \pi(x)\phi_x$, and fact (F1). In addition, values lie in $[0,1]$, and the reference is admissible through the always-reference action: every $\mathcal{U}(x)$ contains the direct call b of the reference, with $v(x,b) = \mathbb{E}[Y_B \mid x]$ and $c(x,b) = \mathbb{E}[C_B \mid x]$. Each action u invokes the reference on an event R_u (R_b is the sure event) and has reference work $r(x,u) = \mathbb{E}[C_B \mathbf{1}_{R_u} \mid x]$; we take $C_B > 0$, so $r(x,u) > 0$ if and only if $\mathbb{P}(R_u \mid x) > 0$. The action is *reference-calling* in x if $\mathbb{P}(R_u \mid x) > 0$ and *reference-free* otherwise. Write $\mathcal{U}^B(x)$ and $\mathcal{U}^0(x)$ for the two sets; we assume $\mathcal{U}^0(x) \neq \emptyset$ in every state. A router is reference-free if every $g(\cdot \mid x)$ is supported on $\mathcal{U}^0(x)$; otherwise it *calls the reference*, since then $\sum_x \pi(x) \sum_u g(u \mid x)\, \mathbb{P}(R_u \mid x) > 0$. Write $Q_g = \mathbb{E}_g[V] - \bar{v}$. For every real ϵ let $\mathcal{G}_\epsilon = \{g \in \mathcal{G} : \mathbb{E}_g[V] \geq \bar{v} - \epsilon\}$. A router is *ratio-optimal* if it maximizes η over $\mathcal{G}$ and *floor-optimal* if it maximizes η over $\mathcal{G}_\epsilon$.

For $\lambda \geq 0$ define the reference-free frontier and the admission band of $u \in \mathcal{U}^B(x)$,

$$F_x(\lambda) = \max_{u \in \mathcal{U}^0(x)} \big[v(x,u) - \lambda c(x,u)\big], \qquad \Lambda_x(u) = \big\{\lambda \geq 0 :\ v(x,u) - \lambda c(x,u) \geq F_x(\lambda)\big\},$$

with $\lambda_x^{\mathrm{lo}}(u) = \inf \Lambda_x(u)$ and $\lambda_x^{\mathrm{hi}}(u) = \sup \Lambda_x(u)$ ($-\infty$ if the band is empty), and the *admission price*

$$\lambda^B = \max\big(\{0\} \cup \{\lambda_x^{\mathrm{hi}}(u) : x \in \mathcal{X},\ u \in \mathcal{U}^B(x)\}\big).$$

If $\lambda^B < \infty$, let λ^+ be the smallest breakpoint of any ϕ_x in (λ^B, ∞), or $\lambda^+ = \infty$ if there is none, and $I^B = (\lambda^B, \lambda^+)$. Let g^B be the deterministic router whose action in state x is, among the maximizers of $v(x,u) - \lambda c(x,u)$ for $\lambda \in I^B$, one of highest value, further ties broken by a fixed order of $\mathcal{U}(x)$. Put

$$\delta^B = Q_{g^B}, \qquad \delta^{\mathrm{free}} = \sum_x \pi(x) \max_{u \in \mathcal{U}^0(x)} v(x,u) - \bar{v}, \qquad \delta^{\max} = \sum_x \pi(x) \max_{u \in \mathcal{U}(x)} v(x,u) - \bar{v}.$$

For a cascade (5) with primary P, $R_u = \{H = 0\}$. We write C_P for the primary's own cost, P also for the action of direct dispatch to P, with $(V, D) = (Y_P, C_P)$, and $\mu_x(u) = \mathbb{E}[(Y_B - Y_P)\mathbf{1}_{R_u} \mid x]/r(x,u)$ and $\eta_B(R_u \mid x) = \mathbb{E}[Y_B \mathbf{1}_{R_u} \mid x]/r(x,u) = \mathbb{E}[Y_B \mid R_u, x]/\mathbb{E}[C_B \mid R_u, x]$.

Two lemmas carry the proof. The first is the extension of Theorem 2(iii) to floors that demand a gain. The objective (2) states the floor for $\epsilon \geq 0$, where the reference itself is feasible; for $\epsilon < 0$ it is not, and feasibility must be assumed.

Lemma 15 (Floors that demand a gain)**.** *For every real ϵ, $\mathcal{G}_\epsilon \neq \emptyset$ if and only if $\epsilon \geq -\delta^{\max}$. For every $\epsilon \geq -\delta^{\max}$, the conclusions of Theorem 2(iii) hold with $\eta_\epsilon^\star = \max_{\mathcal{G}_\epsilon} \eta$: a floor-optimal router exists and can be chosen to randomize in at most one state, and a multiplier $\kappa \geq 0$ exists such that every floor-optimal router uses only actions maximizing $(1+\kappa)v(x,u) - \eta_\epsilon^\star c(x,u)$. Equivalently, every floor-optimal router maximizes J_{λ_ϵ} over $\mathcal{G}$, where $\lambda_\epsilon = \eta_\epsilon^\star/(1+\kappa) \in [0, \eta^\star]$. For every valid κ, either $\kappa = 0$ and $\eta_\epsilon^\star = \eta^\star$, or $\kappa > 0$ and every floor-optimal router has $Q_g = -\epsilon$.*

Proof. By (F1) with $\phi = v$, $\max_{g \in \mathcal{G}} \mathbb{E}_g[V] = \sum_x \pi(x) \max_u v(x,u) = \bar{v} + \delta^{\max}$, which proves the first claim. The proof of Theorem 2(iii) in Appendix H.3 uses the admissibility of the reference, and the sign of ϵ, only to exhibit an element of $\mathcal{G}_\epsilon$. That element is needed four times: for the existence of a constrained maximizer on the compact polytope $\mathcal{G}_\epsilon$; for the feasibility of the Charnes–Cooper program (51), whose slack $\sigma = t(\mathbb{E}_g[V] - \bar{v} + \epsilon)$ is nonnegative exactly because $g \in \mathcal{G}_\epsilon$; for the root of Φ_ϵ; and for the feasibility of (52). Replacing g_B by any element of $\mathcal{G}_\epsilon$, every step holds verbatim for every real $\epsilon \geq -\delta^{\max}$. The multiplier comes from linear-programming strong duality, which needs only a feasible and bounded program and no interior point, so it exists even at $\epsilon = -\delta^{\max}$, where $\mathcal{G}_\epsilon$ is the face of value-maximizing routers. In that proof every floor-optimal router maximizes $f + \kappa h = (1+\kappa)J_{\lambda_\epsilon} + \kappa(\epsilon - \bar{v})$ over $\mathcal{G}$, which is the equivalent form, and $0 \leq \eta_\epsilon^\star \leq \eta^\star$ and $\kappa \geq 0$ give $\lambda_\epsilon \in [0, \eta^\star]$. The last sentence restates the complementary slackness $\kappa(\mathbb{E}_g[V] - \bar{v} + \epsilon) = 0$ and the characterization of $\kappa = 0$ proved there. □

Lemma 16 (Revealed preference along prices)**.** *If g maximizes J_λ and g' maximizes $J_{\lambda'}$ over $\mathcal{G}$ with $0 \leq \lambda < \lambda'$, then $\mathbb{E}_g[D] \geq \mathbb{E}_{g'}[D]$ and $Q_g \geq Q_{g'}$.*

Proof. Adding $J_\lambda(g) \geq J_\lambda(g')$ and $J_{\lambda'}(g') \geq J_{\lambda'}(g)$ gives $(\lambda' - \lambda)(\mathbb{E}_g[D] - \mathbb{E}_{g'}[D]) \geq 0$. Then $Q_g - Q_{g'} = \mathbb{E}_g[V] - \mathbb{E}_{g'}[V] \geq \lambda(\mathbb{E}_g[D] - \mathbb{E}_{g'}[D]) \geq 0$. □

Proposition 4 (full statement). (a) Admission band. *For every x and $u \in \mathcal{U}^B(x)$, $\Lambda_x(u)$ is a closed interval, possibly empty or unbounded above, and u maximizes $v(x,\cdot) - \lambda c(x,\cdot)$ over $\mathcal{U}(x)$ only if $\lambda \in \Lambda_x(u)$. If $\lambda^B < \infty$, every maximizer of J_λ is reference-free for every $\lambda > \lambda^B$, and if some band is nonempty, some reference-calling action maximizes $v(x,\cdot) - \lambda^B c(x,\cdot)$ in some state: λ^B is the largest price at which the reference is admitted. Let u be a cascade* (5) *with primary P, any acceptance rule $H \in \{0,1\}$, possibly randomized and observing anything, and any primary-plus-verification cost $A \geq C_P$, and let direct dispatch to P belong to $\mathcal{U}^0(x)$. If $\Lambda_x(u) \neq \emptyset$, then*

$$\lambda_x^{\mathrm{hi}}(u) \;\leq\; \mu_x(u) \;\leq\; \eta_B(R_u \mid x) \;\leq\; 1/\mathbb{E}[C_B \mid R_u, x].$$

(b) Regimes of the tolerance. *Let $\lambda^B < \eta^\star$. Then g^B is well defined and reference-free, every router that maximizes J_λ for some $\lambda \in I^B$ has quality change δ^B, whatever the tie-break, and $\delta^B \leq \delta^{\mathrm{free}} \leq \delta^{\max}$.*

- *(i)* Redundant. *Every ratio-optimal router is reference-free. For every real $\epsilon > -\delta^B$, $\mathcal{G}_\epsilon \neq \emptyset$, every floor-optimal router is reference-free, and $\lambda_\epsilon > \lambda^B$ for every valid multiplier.*
- *(ii)* Indispensable. *For every real $\epsilon < -\delta^{\mathrm{free}}$, every router in $\mathcal{G}_\epsilon$ calls the reference, and $\mathcal{G}_\epsilon \neq \emptyset$ if and only if $\epsilon \geq -\delta^{\max}$. This part does not use $\lambda^B < \eta^\star$.*
- *(iii)* Purchased. *For every $\epsilon < -\delta^B$ with $\mathcal{G}_\epsilon \neq \emptyset$, every valid multiplier has $\kappa > 0$ and $\lambda_\epsilon \leq \lambda^B$: the floor binds and is priced where reference-calling actions can be admitted. If also $\epsilon \geq -\delta^{\mathrm{free}}$, some reference-free router meets the floor. If $\lambda^B > 0$ and λ^B is not a breakpoint of any F_x, let $Q^- = \max\{Q_g : g \text{ maximizes } J_{\lambda^B}\}$. Then $Q^- > \delta^B$, and for every $\epsilon \in [-Q^-, -\delta^B)$ every floor-optimal router calls the reference.*

(c) Refining the states. *Let a finer state x' determine $x = \rho(x')$, let every action of $\mathcal{U}(x)$ be available in each $x' \in \rho^{-1}(x)$, and let $v(x,u) = \sum_{x' \in \rho^{-1}(x)} \pi(x' \mid x)\, v(x',u)$ and $\mathbb{P}(R_u \mid x) = \sum_{x' \in \rho^{-1}(x)} \pi(x' \mid x)\, \mathbb{P}(R_u \mid x')$, as holds when u acts on a task in the same way whatever its state label. Then δ^{free} and $\delta^{\max}$ do not decrease: refinement can only lower the thresholds $-\delta^{\mathrm{free}}$ and $-\delta^{\max}$, shrinking the indispensable regime and enlarging the feasible range of ϵ. No such monotonicity is asserted for λ^B or δ^B.*

The tie-break toward higher value in the definition of g^B is immaterial: on I^B all maximizers share their value and cost state by state. Between the thresholds the proposition does not say that the reference is always bought. Purchase is guaranteed just below $-\delta^B$, but a reference-free action can retake the frontier at lower prices. One state illustrates both points. Take free actions f_1, f_2, f_3 with $(c,v) = (1, 0.4)$, $(3, 0.7)$ and $(6, 0.8)$, a reference-calling action u with $(2, 0.6)$, and the reference b with $(10, 0.5)$. Then $\Lambda(u) = [0.1, 0.2]$, $\lambda^B = 0.2 < \eta^\star = 0.4$, $\delta^B = -0.1$, $Q^- = 0.1$ and $\delta^{\mathrm{free}} = \delta^{\max} = 0.3$. The floor optimum is f_1 for $\epsilon \geq 0.1$, uses u, alone or mixed with f_1 or f_2, for $-0.2 < \epsilon < 0.1$, and mixes f_2 and f_3 for $-0.3 \leq \epsilon \leq -0.2$. The reference is thus bought only on part of $[-\delta^{\mathrm{free}}, -\delta^B)$, and here the indispensable regime is empty because $\delta^{\mathrm{free}} = \delta^{\max}$.

Part (c) does not extend to λ^B or δ^B: refinement can raise λ^B, lower δ^B, and move the reference from redundant to purchased. Let two equiprobable states share the free actions $f_1 = (1, 0.4)$ and $f_2 = (2, 0.55)$ and the reference $b = (10, 0.525)$, and let the reference-calling action u have $(1.5, 0.525)$ in the first state and $(1.5, 0.4)$ in the second. Pooled, $u = (1.5, 0.4625)$ lies below the chord from f_1 to f_2, so every band is empty, $\lambda^B = 0$, $g^B = f_2$ and $\delta^B = +0.025$: the reference is redundant at $\epsilon = 0$. Refined, $\Lambda(u) = [0.05, 0.25]$ in the first state, so $\lambda^B = 0.25 < \eta^\star = 0.4$ and $\delta^B = -0.125$, while $\delta^{\mathrm{free}} = +0.025$ is unchanged. The $\epsilon = 0$ floor optimum now plays u in the first state, with efficiency 0.315 against 0.286 for the best reference-free router, so the reference is purchased.

Proof. Part (a). F_x is a maximum of finitely many affine functions, hence convex and continuous. So $\lambda \mapsto v(x,u) - \lambda c(x,u) - F_x(\lambda)$ is concave and continuous, its superlevel set at 0 is a closed interval, and so is its intersection $\Lambda_x(u)$ with $[0, \infty)$. Since $\mathcal{U}^0(x) \subseteq \mathcal{U}(x)$, an action that maximizes over $\mathcal{U}(x)$ at λ satisfies $v(x,u) - \lambda c(x,u) \geq F_x(\lambda)$, that is, $\lambda \in \Lambda_x(u)$. If $\lambda > \lambda^B$, then $\lambda \notin \Lambda_x(u)$ for every reference-calling u, so $v(x,u) - \lambda c(x,u) < F_x(\lambda) \leq \phi_x(\lambda)$; no reference-calling action maximizes, and by (F1) every maximizer of J_λ is reference-free. If some band is nonempty, there are finitely many pairs and the bands are closed, so $\lambda^B = \lambda_{x^*}^{\mathrm{hi}}(u^*) \in \Lambda_{x^*}(u^*)$ for some pair. Were

some $u' \in \mathcal{U}(x^*)$ strictly better than u^* at λ^B, it would exceed $F_{x^*}(\lambda^B)$, hence be reference-calling, and by continuity it would exceed F_{x^*} on a right neighbourhood of λ^B, giving $\lambda^{\mathrm{hi}}_{x^*}(u') > \lambda^B$. So u^* maximizes at λ^B.

For the cascade, $V = HY_P + (1-H)Y_B$ and $D = A + (1-H)C_B$, while direct dispatch to P has $(V, D) = (Y_P, C_P)$. Hence

$$v(x,u) - v(x,P) = \mathbb{E}[(Y_B - Y_P)\mathbf{1}_{R_u} \mid x] =: N,$$
$$c(x,u) - c(x,P) = \mathbb{E}[A - C_P \mid x] + r(x,u) \;\geq\; r(x,u) > 0.$$

If $\lambda \in \Lambda_x(u)$, then $v(x,u) - \lambda c(x,u) \geq F_x(\lambda) \geq v(x,P) - \lambda c(x,P)$, so $\lambda\,[c(x,u) - c(x,P)] \leq N$. A nonempty band therefore forces $N \geq 0$, and then $\lambda \leq N/[c(x,u) - c(x,P)] \leq N/r(x,u) = \mu_x(u)$; taking the supremum gives $\lambda^{\mathrm{hi}}_x(u) \leq \mu_x(u)$. Next, $Y_P \geq 0$ gives $N \leq \mathbb{E}[Y_B \mathbf{1}_{R_u} \mid x]$, so $\mu_x(u) \leq \eta_B(R_u \mid x)$. Finally $Y_B \leq 1$ gives $\mathbb{E}[Y_B \mathbf{1}_{R_u} \mid x] \leq \mathbb{P}(R_u \mid x)$, so $\eta_B(R_u \mid x) \leq \mathbb{P}(R_u \mid x)/r(x,u) = 1/\mathbb{E}[C_B \mid R_u, x]$. Only $H \in \{0,1\}$, $A \geq C_P$ and direct dispatch to P, which links F_x to μ_x, were used, so the bound holds for every verifier and verifier cost. Without direct dispatch to P it can fail: with $Y_P = 0$, $A = C_P = 1$, $Y_B = C_B = 1$ and $H = 0$, the cascade has $(c, v) = (2, 1)$ and $\mu = 1$, and against a single reference-free action $(10, 0)$ its band is $[0, \infty)$.

Part (b), preliminaries. Since $\lambda^B < \eta^\star < \infty$, I^B is a nonempty open interval. No ϕ_x has a breakpoint on I^B, so $\phi_x(\lambda) = a_x - b_x\lambda$ there. For $u \in \mathcal{U}(x)$ the difference $\phi_x(\lambda) - [v(x,u) - \lambda c(x,u)]$ is affine and nonnegative on I^B; if it vanishes at one point of the open interval, it vanishes on all of it. So the maximizer set M_x is the same for every $\lambda \in I^B$, and every element of M_x has $(v, c) = (a_x, b_x)$. Hence g^B is well defined, and by (F1) every router that maximizes J_λ for some $\lambda \in I^B$ has value $\sum_x \pi(x) a_x$ and cost $\sum_x \pi(x) b_x$, whatever the tie-break. In particular its quality change is δ^B. By continuity, g^B maximizes J_λ on the closure of I^B. By part (a), $M_x \subseteq \mathcal{U}^0(x)$, so g^B is reference-free. A reference-free router has $\mathbb{E}_g[V] \leq \sum_x \pi(x) \max_{\mathcal{U}^0(x)} v = \bar{v} + \delta^{\mathrm{free}}$, so $\delta^B \leq \delta^{\mathrm{free}}$, and $\delta^{\mathrm{free}} \leq \delta^{\max}$ because $\mathcal{U}^0(x) \subseteq \mathcal{U}(x)$. We also use:

$(*)$ If g maximizes J_λ for some $\lambda > \lambda^B$, then $Q_g \leq \delta^B$.

For $\lambda \in I^B$, $Q_g = \delta^B$. For $\lambda \geq \lambda^+$, apply Lemma 16 to g^B at some $\lambda' \in I^B$, $\lambda' < \lambda$.

Part (b)(i). By Theorem 2(i), ratio-optimal routers use only maximizers of $v - \eta^\star c$, which are reference-free by part (a) because $\eta^\star > \lambda^B$. Let $\epsilon > -\delta^B$. Then $Q_{g^B} = \delta^B > -\epsilon$, so $g^B \in \mathcal{G}_\epsilon$ and Lemma 15 applies. Let g be floor-optimal and κ a valid multiplier. If $\kappa = 0$, then $\lambda_\epsilon = \eta^\star > \lambda^B$. If $\kappa > 0$, then $Q_g = -\epsilon < \delta^B$. Were $\lambda_\epsilon \leq \lambda^B$, Lemma 16, applied to g at λ_ϵ and to g^B at any $\lambda' \in I^B$, would give $Q_g \geq \delta^B$; hence again $\lambda_\epsilon > \lambda^B$. In both cases g maximizes J_{λ_ϵ} by Lemma 15, and it is reference-free by part (a).

Part (b)(ii). A reference-free router has $Q_g \leq \delta^{\mathrm{free}}$ (see the preliminaries). If $\epsilon < -\delta^{\mathrm{free}}$, a router in $\mathcal{G}_\epsilon$ has $Q_g \geq -\epsilon > \delta^{\mathrm{free}}$, so it is not reference-free: it puts positive mass on some $u \in \mathcal{U}^B(x)$ in some state, and since $\pi(x) > 0$ and $\mathbb{P}(R_u \mid x) > 0$, it calls the reference. Nonemptiness is Lemma 15.

Part (b)(iii). Let $\epsilon < -\delta^B$, $\mathcal{G}_\epsilon \neq \emptyset$, g floor-optimal and κ a valid multiplier. Ratio-optimal routers maximize $J_{\eta^\star}$ with $\eta^\star > \lambda^B$, so by $(*)$ they have $Q \leq \delta^B < -\epsilon$ and none meets the floor. By Lemma 15, $\kappa = 0$ is not valid, so $\kappa > 0$ and $Q_g = -\epsilon$. If $\lambda_\epsilon > \lambda^B$, then $(*)$ gives $Q_g \leq \delta^B < -\epsilon$, a contradiction; hence $\lambda_\epsilon \leq \lambda^B$. If $\epsilon \geq -\delta^{\mathrm{free}}$, the reference-free router that plays a value-maximizing action of $\mathcal{U}^0(x)$ in every state has $Q = \delta^{\mathrm{free}} \geq -\epsilon$.

For the purchase window, let $\lambda^B > 0$ be attained by (x^*, u^*), so that u^* maximizes at λ^B by part (a), and let $f = g^B(x^*)$. Both attain $\phi_{x^*}(\lambda^B)$, f by continuity. Since f maximizes for $\lambda > \lambda^B$ and u^* does not, $c(x^*, u^*) > c(x^*, f)$, and therefore $v(x^*, u^*) - v(x^*, f) = \lambda^B[c(x^*, u^*) - c(x^*, f)] > 0$. The router that equals g^B except that it plays u^* in x^* maximizes J_{λ^B} by (F1), and its quality change is $\delta^B + \pi(x^*)[v(x^*, u^*) - v(x^*, f)] > \delta^B$. The maximizers of J_{λ^B} form a compact face of $\mathcal{G}$, so Q^- is attained and $Q^- > \delta^B$. Now let $\epsilon \in [-Q^-, -\delta^B)$. Then $\mathcal{G}_\epsilon \neq \emptyset$, and by the first paragraph a floor-optimal g has $Q_g = -\epsilon \in (\delta^B, Q^-]$ and $\lambda_\epsilon \leq \lambda^B$.

We first show that g maximizes J_{λ^B}. This is immediate if $\lambda_\epsilon = \lambda^B$. If $\lambda_\epsilon < \lambda^B$, let g^- maximize J_{λ^B} with $Q_{g^-} = Q^-$. Lemma 16 gives $\mathbb{E}_g[D] \geq \mathbb{E}_{g^-}[D]$ and $Q_g \geq Q^- \geq -\epsilon = Q_g$, so $Q_g = Q^-$. Optimality of g at λ_ϵ then gives $0 = Q_g - Q^- \geq \lambda_\epsilon(\mathbb{E}_g[D] - \mathbb{E}_{g^-}[D]) \geq 0$. The floor value

$\bar{v} - \epsilon > \bar{v} + \delta^B = \mathbb{E}_{g^B}[V] \geq 0$ is positive, so $\eta^\star_\epsilon > 0$ and $\lambda_\epsilon > 0$. Hence $\mathbb{E}_g[D] = \mathbb{E}_{g^-}[D]$, and so $J_{\lambda^B}(g) = J_{\lambda^B}(g^-) = \Phi(\lambda^B)$.

Suppose now that g were reference-free. By (F1), in every state g uses only actions of $\mathcal{U}^0(x)$ that attain $\phi_x(\lambda^B)$, so these actions attain $F_x(\lambda^B)$. Since F_x has no breakpoint at λ^B, it is affine on a neighbourhood of λ^B, and an action attaining $F_x(\lambda^B)$ attains F_x on that whole neighbourhood, by the argument of the preliminaries. For $\lambda > \lambda^B$ every maximizer is reference-free, so $F_x = \phi_x$ there. Hence the action maximizes on a right neighbourhood of λ^B inside I^B, and its value and cost are (a_x, b_x). Then $Q_g = \delta^B$, which contradicts $Q_g > \delta^B$. So g calls the reference.

Part (c). If u is reference-free in x, then $0 = \mathbb{P}(R_u \mid x) = \sum_{x'} \pi(x' \mid x)\, \mathbb{P}(R_u \mid x')$ makes it reference-free in every $x' \in \rho^{-1}(x)$, so $\mathcal{U}^0(x) \subseteq \mathcal{U}^0(x')$. Hence

$$\begin{aligned}\sum_{x'} \pi(x') \max_{u \in \mathcal{U}^0(x')} v(x', u) &\geq \sum_{x} \pi(x) \sum_{x' \in \rho^{-1}(x)} \pi(x' \mid x) \max_{u \in \mathcal{U}^0(x)} v(x', u) \\ &\geq \sum_{x} \pi(x) \max_{u \in \mathcal{U}^0(x)} v(x, u),\end{aligned}$$

the second step because an average of maxima is at least the maximum of the average. The reference mean $\bar{v}$ does not depend on the partition, so δ^{free} does not decrease. The same argument with $\mathcal{U}(x) \subseteq \mathcal{U}(x')$ applies to $\delta^{\max}$. □

A reference-price threshold. The admission price also bounds how far the reference's price must fall before it is bought. Split each cost as $c(x,u) = c^o(x,u) + r(x,u)$, where c^o does not pay the reference, and scale the reference's price by $t > 0$, so that $c_t = c^o + t\, r$ and values are unchanged. Reference-free actions have $r = 0$, so their costs, F_x and δ^{free} do not depend on t.

Corollary 17 (Reference-price threshold, a sufficient condition)**.** *Fix $\epsilon \geq -\delta^{\text{free}}$, or $\epsilon = +\infty$ for the ratio optimum. Let η^0_ϵ and κ_0 be the optimum and a multiplier of the floor problem restricted to reference-free routers (Lemma 15 on the menus $\mathcal{U}^0(x)$; $\kappa_0 = 0$ if $\epsilon = +\infty$). Assume $\lambda_0 = \eta^0_\epsilon/(1+\kappa_0) > 0$, and put*

$$t^\star_\epsilon = \max_{x,\ u \in \mathcal{U}^B(x)} \frac{v(x,u) - \lambda_0 c^o(x,u) - F_x(\lambda_0)}{\lambda_0\, r(x,u)}.$$

If $t > t^\star_\epsilon$, then at reference price t every floor-optimal router is reference-free and the optimum is η^0_ϵ. For a cascade as in part (a), the term of u is at most $\mu_x(u)/\lambda_0$, with μ evaluated at $t = 1$.

Proof. Write $f_0(g) = \mathbb{E}_g[V - \eta^0_\epsilon D]$ and $h(g) = \mathbb{E}_g[V] - \bar{v} + \epsilon$; if $\epsilon = +\infty$, drop every term that contains h or κ_0. For the restricted problem, Lemma 15 and the strong duality in its proof give $\max_g [f_0(g) + \kappa_0 h(g)] = 0$ over reference-free routers. By (F1) that maximum equals $(1+\kappa_0)\sum_x \pi(x) F_x(\lambda_0) + \kappa_0(\epsilon - \bar{v})$. For $t > t^\star_\epsilon$, every reference-calling u has $v(x,u) - \lambda_0 c_t(x,u) < F_x(\lambda_0)$. So for every router $g \in \mathcal{G}$, (F1) gives

$$f_0(g) + \kappa_0 h(g) = (1+\kappa_0) J_{\lambda_0}(g) + \kappa_0(\epsilon - \bar{v}) \leq 0,$$

with strict inequality if g calls the reference. If $g \in \mathcal{G}_\epsilon$, then $h(g) \geq 0$, so $\mathbb{E}_g[V - \eta^0_\epsilon D] \leq 0$ and $\eta(g) \leq \eta^0_\epsilon$ by (49). The restricted optimum attains η^0_ϵ. If $\eta(g) = \eta^0_\epsilon$, then $f_0(g) = 0$ and $f_0(g) + \kappa_0 h(g) \geq 0$, so equality holds and g is reference-free. For the cascade bound, $F_x(\lambda_0) \geq v(x,P) - \lambda_0 c(x,P)$ and $c^o(x,u) = \mathbb{E}[A \mid x] \geq c(x,P)$. So the numerator is at most $v(x,u) - v(x,P) = \mu_x(u)\, r(x,u)$. □

The threshold is sufficient only. Below $t^\star_\epsilon$ the certificate fails, but reference-free routers can remain optimal, for instance under another valid κ_0. Where the reference actually enters is therefore an empirical question, answered below by refits.

Remark 18 (Relation to cascade routing and to fractional programming). Dekoninck et al. (2025) maximize estimated quality under a cost budget through a Lagrangian price and derive routing and cascading rules at that price. The reference has no distinguished role there, and the price is tuned to the budget. Proposition 4 differs in three ways. First, the price is not free: the reference and the tolerance fix it through Lemma 15. The question is whether the action that defines the yardstick is also worth buying as an input. Second, part (a) needs no estimator: whenever the primary can also

be dispatched directly and $A \geq C_P$, the price at which a cascade can be admitted is bounded by the reference's own efficiency on the tasks it would receive, for every verifier. Third, the regimes are indexed by the tolerance, including floors that demand a gain, which a budget does not express. In classical terms, Dinkelbach's parametrization and the Charnes–Cooper program (Dinkelbach, 1967; Charnes & Cooper, 1962) reduce the problem to linear programs whose objective is parametric in the price λ, the setting of Gass & Saaty (1955). The band $\Lambda_x(u)$ is the price range over which the column of u is optimal against the reference-free columns, and parts (b)(i)–(iii) add a parametric analysis in the right-hand side $\bar{v} - \epsilon$ of the floor, which that algorithm does not treat. What is specific is that one object appears both as a column of the program and in that right-hand side. The regime boundaries depend on the two roles only through three scalars, λ^B, δ^B and δ^{free}; whether the reference is bought inside the middle regime depends on more, and is guaranteed only on $[-Q^-, -\delta^B)$.

Empirical check. All quantities are fitted on the 7,080 probing tasks with the 22-action menu of Section 3.1, whose cascades have $A = C_P$ and oracle acceptance and whose primaries are all available by direct dispatch; $\eta_B = 34.73$ score per dollar on probing. *Part (a).* On the 21 cells, $\mu_x(u) \leq \eta_B(R_u \mid x) \leq 1/\mathbb{E}[C_B \mid R_u, x]$ holds in all 278 state–cascade pairs (126 on nine dimensions), and $\lambda_x^{\text{hi}}(u) \leq \mu_x(u)$ holds for every admitted cascade. The comparison with the reference's efficiency is per state, not global. Per-cell μ exceeds the global η_B in 23% of the pairs and reaches $14.2\eta_B$ (qwen3.5-plus at $\theta = 1$ on code_generation|humaneval). In 69% of the pairs $\eta_B(R_u \mid x) < \eta_B(x)$ (median ratio 0.68): the tasks a primary fails are also harder or costlier for Opus. We therefore report pooled returns. Pooled over tasks, a Qwen3-Max fallback dollar returns $\mu = 0.41\eta_B$ on probing at $\theta = 1$ ($0.39\eta_B$ $[0.35, 0.42]$ on the test set), and $0.44\eta_B$ at $\theta = 0.5$ ($0.43\eta_B$ $[0.39, 0.47]$). Without states, the band of Qwen3-Max at $\theta = 0.5$ ends exactly at its $\mu = 0.441\eta_B$, which is the pooled λ^B. There the bound is tight, because the verifier is free and direct Qwen3-Max is the binding alternative.

Part (b). Table 22 gives the regime of each state scheme. On every scheme $\eta^\star > \lambda^B$, and λ^B is not a breakpoint of any F_x, so part (b)(iii) applies. LP refits of the floor across demanded gains $-\epsilon$ confirm the switch. We report two shares: the share of tasks sent to a reference-calling action, and the smaller share on which Opus runs, $\sum_x \pi(x) \sum_u g(u \mid x)\, \mathbb{P}(R_u \mid x)$, since a cascade calls Opus only on the tasks it rejects. On the 21 cells, no refit with $-\epsilon \leq \delta^B - 10^{-4}$ calls Opus. The refit at $-\epsilon = \delta^B + 10^{-4}$ does, at price exactly λ^B and inside the purchase window $\epsilon \in [-Q^-, -\delta^B) = [-0.069, -0.056)$: it sends 0.06% of tasks to the kimi-k2.5 ($\theta = 0.5$) cascade, and Opus runs on 0.01%. At $-\epsilon = \delta^B + 0.005$ the refit sends 3.2% of tasks to that cascade, Opus runs on 0.58% of probing and 0.62% of test tasks, and on the test set it gives $14.4\times$ with $\Delta V = +0.046$ $[0.034, 0.059]$. At $-\epsilon = 0.08 > \delta^{\text{free}}$ the reference is indispensable; at price $1.1\eta_B$ the refit sends 9.2% of tasks to two cascades, Opus runs on 3.1% of probing and 2.7% of test tasks, and it gives $11.9\times$ with $+0.061$ $[0.049, 0.073]$. Nine dimensions switch at $\delta^B = 0.046$ in the same way; Opus runs on 0.01%, 0.71% and 4.2% of probing tasks at the three refits. In 1,000 probing bootstraps with every quantity refitted, $\lambda^B < \lambda_0$ in 96.8% of resamples on the 21 cells and in 100% on nine dimensions. The 95% interval of δ^B is $[-0.002, +0.066]$ on the 21 cells, which includes zero, and $[0.039, 0.057]$ on nine dimensions. Fitted on the test set itself, the 21 cells give $\lambda^B = 2.42\eta_B < \lambda_0 = 4.07\eta_B$ in units of the test-set $\eta_B = 37.6$, and $\delta^B = +0.026$.

Part (c) and the price threshold. Consistent with part (c), δ^{free} rises from -0.020 without states to $+0.077$ on nine dimensions and $+0.079$ on 21 cells; the per-task in-sample ceiling on the test set is $+0.147$. By contrast, δ^B carries no such guarantee. It happens to rise along the nested true partitions (-0.045, $+0.046$, $+0.056$). It is -0.006 on predicted cells against $+0.053$ on predicted dimensions, but these come from separate classifiers and are not nested; the example after the full statement shows that refinement itself can lower δ^B. For Corollary 17 on the 21 cells, $t_0^\star = 0.693$: Opus would have to become at least 1.44 times cheaper before the zero-loss floor can call it. Refits confirm the threshold rather than prove it. The refit at $t = (1 - 10^{-3})t_0^\star$ calls Opus, and none at $t = (1 + 10^{-6})t_0^\star$ or at the grid prices above $t_0^\star$ does. For the ratio optimum, $t^\star = 0.105$, a 9.6-fold drop, and the first grid refit that calls Opus is at $t = 0.104$. On nine dimensions the values are 0.570 and 0.099, with first refits at 0.570 and 0.096.

Every row agrees with the proposition. In the redundant rows $\lambda_0 > \lambda^B$, as part (b)(i) requires. In the purchased and indispensable rows $\lambda_0 \leq \lambda^B$, as part (b)(iii) requires. Without states the zero-loss policy sends 44.9% of tasks to the Qwen3-Max ($\theta = 0.5$) cascade and the rest to direct Qwen3-Max, so Opus runs on 28.7% of probing and 28.3% of test tasks. On the test set it gives

Table 22: **Regimes of the reference on the policy-study menu.** Fitted on probing: $\eta^\star$, the admission price λ^B and the zero-loss floor price $\lambda_0 = \eta_0^\star/(1+\kappa)$, in units of η_B, and δ^B, δ^{free}, $\delta^{\max}$. At $\epsilon = 0$ the reference is redundant if $\delta^B > 0$, purchased if $\delta^{\text{free}} \geq 0 > \delta^B$, and indispensable if $\delta^{\text{free}} < 0$; $\delta^B = 0$ is a boundary case that the proposition does not cover. "To B" is the fitted share of tasks that the $\epsilon = 0$ policy sends to a reference-calling action, and "Opus" the fitted share on which Opus runs, $\sum_x \pi(x) \sum_u g(u \mid x)\, \mathbb{P}(R_u \mid x)$, since a cascade calls Opus only on the tasks it rejects. The last two columns evaluate that policy on the 2,919 test tasks. Predicted (Pred.) cells and dimensions (dim.) come from the prompt-only classifiers of Section 3.1.

	Prices ($\times\eta_B$)							Share		Test	
States	$\eta^\star$	λ^B	λ_0	δ^B	δ^{free}	$\delta^{\max}$	Regime	To B	Opus	η/η_B	ΔV
None (pooled)	20.85	0.44	0.44	-0.045	-0.020	0.070	indispensable	44.9%	28.7%	3.62	$+0.0021$
Nine dim.	28.86	2.26	3.96	$+0.046$	$+0.077$	0.117	redundant	0	0	22.73	-0.0090
21 cells	29.42	2.88	4.16	$+0.056$	$+0.079$	0.118	redundant	0	0	25.52	-0.0044
Pred. dim.	28.93	2.53	4.03	$+0.053$	$+0.076$	0.116	redundant	0	0	24.40	-0.0048
Pred. cells	29.36	7.94	4.15	-0.006	$+0.079$	0.118	purchased	0.5%	0.06%	25.32	-0.0070

$3.62\times$ $[3.41, 3.85]$ and 3.23 m$ per task $[3.05, 3.42]$, with $\Delta V = +0.0021$ $[-0.0085, +0.0135]$. On predicted cells $\epsilon = 0$ lies in the purchased regime but outside the guaranteed window $\epsilon \in [-Q^-, -\delta^B) = [0.0053, 0.0058)$. The fitted policy still sends one predicted cell, code_completion|he with 0.5% of tasks, to the Qwen3-Max ($\theta = 1$) cascade, so Opus runs on 0.06% of probing tasks and on none of the test tasks.

H.5 Proof of Proposition 9

Proof. Subtract informed cost from each uninformed alternative:

$$C_{\text{direct}}(s) - C_{\text{info}}(s) = s(b-a) - k, \qquad C_{\text{cascade}}(s) - C_{\text{info}}(s) = a(1-s) - k.$$

Both differences are strictly positive if and only if $k/(b-a) < s < 1 - k/a$, which is (28). The interval is nonempty if and only if

$$\frac{k}{b-a} < 1 - \frac{k}{a} \iff k < \frac{a(b-a)}{b}.$$

This same condition places a/b strictly between the two endpoints. Since $k \geq 0$, the band lies within $0 < s < 1$. Equality at an endpoint means that information ties one uninformed alternative. □

H.6 Imperfect information and the perfect-information bound

Let U denote the binary sufficiency event, with $\mathbb{P}(U = 1) = s$. An experiment Z produces posterior $s_Z = \mathbb{P}(U = 1 \mid Z)$. Put

$$f(s) = \min\{a + (1-s)b, b\}, \qquad g(s) = sa + (1-s)b.$$

After paying for Z, choosing the cheaper available action has expected cost $k + \mathbb{E}[f(s_Z)]$. Thus its net saving is $f(s) - \mathbb{E}[f(s_Z)] - k$.

The minimum of two affine functions is concave. Since $\mathbb{E}[s_Z] = s$, Jensen's inequality gives $\mathbb{E}[f(s_Z)] \leq f(s)$. Also $f(t) \geq g(t)$ for every $t \in [0, 1]$: cascade minus $g(t)$ is $a(1-t)$, and direct minus $g(t)$ is $t(b-a)$. Since g is affine,

$$g(s) = \mathbb{E}[g(s_Z)] \leq \mathbb{E}[f(s_Z)] \leq f(s).$$

Gross information value is therefore at most $f(s) - g(s)$. Exact revelation attains this bound because $f(0) = g(0) = b$ and $f(1) = g(1) = a$. This proves directly the relevant perfect-information comparison (Blackwell, 1953). In particular,

$$f(s) - g(s) = \min\{s(b-a), a(1-s)\},$$

maximized at $s = a/b$ with value $a(b-a)/b$. At the same experiment cost k, no imperfect experiment strictly reduces cost outside the perfect-information band. Experiments with different costs require separate comparisons. In the language of decision theory, $f(s) - g(s)$ is the expected value of perfect information and $f(s) - \mathbb{E}[f(s_Z)]$ the expected value of the experiment Z (Raiffa & Schlaifer, 1961; Howard, 1966), here for two actions and a binary state.

Information can pay only if $k < a$. Exact information saves $a(1-s)-k$ against cheap-first execution, so it can pay only if $k < a(1-s) < a$: it avoids at most one unsuccessful primary attempt per task.

Corollary 19 (Routing information versus a better proposer)**.** *Let a refinement raise sufficiency from s to $s+\Delta p \le 1$ at cost k, so that the refined cascade costs $C_{\mathrm{dec}} = k + a + (1-s-\Delta p)b$, and let exact routing information cost the same k. Then*

$$C_{\mathrm{info}} - C_{\mathrm{dec}} = b\,\Delta p - a(1-s).$$

At equal cost, paying for the routing signal beats improving the proposer if and only if $b\,\Delta p < a(1-s)$.

Proof. $C_{\mathrm{info}} - C_{\mathrm{dec}} = [k + sa + (1-s)b] - [k + a + (1-s)b - \Delta p\, b] = -a(1-s) + b\,\Delta p.$ □

Information avoids wasted attempts, worth a on each of the $1-s$ failing tasks; a better proposer avoids fallbacks, worth b on each of the Δp newly solved tasks. Both comparisons hold the reference fixed; with a correlated reference, the b in $b\,\Delta p$ becomes the reference's mean cost on the newly solved tasks, which lie in the original miss set.

A score used at the wrong threshold can be worse than no signal. Equation (29) values the rule that thresholds the true posterior s_Z at a/b. A learned score $\hat{s}$, a function of Z, used to attempt if and only if $\hat{s} \ge a/b$, has expected cost $k + \mathbb{E}[C_{\alpha(\hat{s})}(s_Z)]$, where $C_\alpha(t)$ is the cost of action α when the posterior is t. Since $C_\alpha(t) \ge \min\{a + (1-t)b,\, b\} = f(t)$ for both actions,

$$k + \mathbb{E}\big[C_{\alpha(\hat{s})}(s_Z)\big] \;\ge\; k + \mathbb{E}[f(s_Z)],$$

with equality if and only if the rule attempts on $\{s_Z > a/b\}$, up to ties. Using the true posterior is sufficient; a monotone transform with the correspondingly transformed threshold also attains the bound. For an incorrect-threshold example, let $a = 1/10$, $b = 1$, $s = 1/2$ and $k = 0$. The uninformed cascade costs $3/5$. Equiprobable posteriors $s_Z \in \{1/20, 19/20\}$ reduce Bayes cost to $23/40$, but reporting $\hat{s} = s_Z/10$ and retaining threshold a/b rejects both, costing 1. A free informative signal can therefore worsen decisions when its score is used at the wrong threshold.

I Verification and Parallel Resource Accounting

I.1 What conditional acceptance reliability guarantees

Let Y indicate primary correctness and let $\mathcal{G}$ contain all information used for commitment, including controller randomization. Suppose $v = \mathbb{E}[Y \mid \mathcal{G}]$ and the measurable event $\{H = 1\}$ is contained in $\{v \ge \tau\}$ for a reliability level $\tau \in [0,1]$. For a positive-probability acceptance event,

$$\mathbb{P}(Y = 1 \mid H = 1) = \mathbb{E}[v \mid H = 1] \ge \tau.$$

For a nonnegative integrable $\mathcal{G}$-measurable stake w,

$$\mathbb{E}[wY \mid H = 1] = \mathbb{E}[wv \mid H = 1] \ge \tau \mathbb{E}[w \mid H = 1].$$

These conclusions require a conditional probability bound on the population selected for acceptance. Reliability measured only for a score in aggregate can change after selection using additional predictive information. The score cutoff θ in $H = \mathbf{1}\{\Sigma \ge \theta\}$ is not a probability guarantee; equating it with τ requires establishing the corresponding conditional reliability. Lemma 1 does not assume such a bound.

I.2 Concurrent (hedged) execution: latency and cancellation charges

Concurrent (hedged) execution launches the primary and the reference at the same time, after controller overhead ℓ_κ, and cancels the reference when the primary is accepted; below we say concurrent execution. Suppose deterministic completion times satisfy $0 < \ell_p \le \ell_b$, with primary verification included in ℓ_p, and the controller waits for the primary verdict. It terminates at $\ell_\kappa + \ell_p$

on a hit and $\ell_\kappa + \ell_b$ on a miss. With hit value v_h and a reference-restoring fallback, so that the miss value is $\bar{v} > 0$,

$$\eta^{\ell}_{\text{hedge}} = \frac{pv_h + (1-p)\bar{v}}{\ell_\kappa + p\ell_p + (1-p)\ell_b},$$
$$\eta^{\ell}_{\text{hedge}} \geq \frac{\bar{v}}{\ell_b} \iff p(\ell_b v_h - \bar{v}\ell_p) \geq \bar{v}\ell_\kappa. \tag{53}$$

This follows by multiplying the two positive denominators and canceling the common miss term. If $\ell_\kappa = 0$, $v_h \geq \bar{v}$, and $\ell_p \leq \ell_b$, the comparison holds, strictly when $p > 0$ and either component comparison is strict. Sequential escalation instead has mean latency $\ell_\kappa + \ell_p + (1-p)\ell_b$ and condition $p\ell_b v_h \geq \bar{v}(\ell_\kappa + \ell_p)$.

For dollars, let $a > 0$ be the primary and controller cost, $b > 0$ the full reference cost, and r the conditional expected charge for a prefix consumed before cancellation on a hit. Then

$$\eta^{c}_{\text{hedge}} = \frac{pv_h + (1-p)\bar{v}}{a + (1-p)b + pr}, \qquad \eta^{c}_{\text{hedge}} \geq \frac{\bar{v}}{b} \iff p(bv_h - \bar{v}r) \geq \bar{v}a. \tag{54}$$

This is a verified, wait-for-verdict variant of the hedged requests of Dean & Barroso (2013), with billing for the consumed fallback prefix. Relative to sequential escalation with the same value, it shortens mean latency by $(1-p)\ell_p$ and raises expenditure by pr.

For random completion times the two resources behave differently. Dollars add across calls, so the constants a, b and r in (54) may be read as conditional expected charges without further change. Latency cannot be handled this way, because the controller waits for the primary verdict and, on a miss, for the fallback as well: the round ends at $\ell_\kappa + L_p$ on a hit and at $\ell_\kappa + \max(L_p, L_b)$ on a miss, and in general $\mathbb{E}[\max(L_p, L_b)] \neq \max(\mathbb{E}L_p, \mathbb{E}L_b)$. With $m_h = \mathbb{E}[L_p \mid H = 1]$, $m_m = \mathbb{E}[\max(L_p, L_b) \mid H = 0]$ and reference latency $l_b = \mathbb{E}[L_b]$, the mean latency of concurrent execution is $\ell_\kappa + pm_h + (1-p)m_m$ and

$$\eta^{\ell}_{\text{hedge}} \geq \frac{\bar{v}}{l_b} \iff p\,(l_b v_h - \bar{v}m_h) \geq \bar{v}\ell_\kappa + (1-p)\,\bar{v}\,(m_m - l_b), \tag{55}$$

which is (53) plus the miss-branch term $(1-p)\bar{v}(m_m - l_b)$; the two agree when $m_m = l_b$, as in the deterministic case $\ell_p \leq \ell_b$. Substituting the mean $\mathbb{E}[L_p]$ for ℓ_p is not valid in general. Take $L_p \in \{1/2, 3/2\}$ equally likely and independent of H, $L_b = 1$, $\ell_\kappa = 0$, $v_h = \bar{v} = 1$ and $p = 1/2$. The mean-substituted criterion reads $0 \geq 0$, whereas $m_h = 1$, $m_m = 5/4$, the mean latency is $9/8 > 1$ and the efficiency is $8/9 < 1$; criterion (55) correctly fails, since $0 \geq 1/8$ is false. Lemma 1 applies verbatim to the latency resource with $A = \ell_\kappa + L_p$, $C_F = (L_b - L_p)^+$ and $C_B = L_b$; what fails is only the substitution of unconditional means for the branch-conditional ones.

J Coverage Allocation Across Independent Positions

J.1 Setting and statement

Suppose K independent positions or task families have increasing, strictly concave differentiable coverage curves $p_k(F_k)$, where $F_k \geq 0$ is a continuous allocation and $\gamma_k > 0$ its unit price. Let $w_k > 0$ be the value of coverage at position k. For J_λ, this weight can include the family frequency multiplied by the margin W_λ in (26); families with nonpositive margin are not attempted first and are excluded, and a positive-margin family must also clear the activation threshold of Appendix J.3. A hard-budget subproblem is

$$\max_{F \geq 0} \sum_{k=1}^{K} w_k p_k(F_k) \quad \text{subject to} \quad \sum_k \gamma_k F_k \leq B. \tag{56}$$

This is separable allocation across positions, not a model of mutually redundant proposers within one position.

Proposition 20 (Coverage allocation)**.** *For $B > 0$,* (56) *has a unique optimizer. There is a multiplier $\Lambda > 0$ such that funded positions satisfy $w_k p_k'(F_k) = \Lambda\gamma_k$, and unfunded positions satisfy $w_k p_k'(0) \leq$*

$\Lambda\gamma_k$. *If* $1 - p_k(F) = d_k(1+F)^{-r}$ *with* $d_k \in (0,1]$ *and common* $r > 0$*, then*

$$F_k^\star = \left[\mu\left(\frac{w_k d_k}{\gamma_k}\right)^{1/(1+r)} - 1\right]_+, \tag{57}$$

where μ *makes the budget bind.*

The exponent depends on the capacity unit. For clean Zipf families, Proposition 13 gives $r = 1 - 1/z$ for a retention horizon or event FIFO length, yielding large-allocation proportions $(w_k d_k/\gamma_k)^{z/(2z-1)}$. For distinct stored records, Corollary 27 gives $r = z - 1$ and proportions $(w_k d_k/\gamma_k)^{1/z}$. These proportions are asymptotic.

J.2 Proof of Proposition 20

Proof. The feasible region $\{F \geq 0\colon \sum_k \gamma_k F_k \leq B\}$ is nonempty, convex, and compact. A maximizer exists by continuity. Positive weights and strict concavity of each separate p_k make the sum strictly concave, so the maximizer is unique.

The budget binds: otherwise some coordinate could be increased feasibly and improve the objective. Since $B > 0$, a strictly feasible point with every coordinate positive exists. The concave problem therefore satisfies the Karush-Kuhn-Tucker conditions, which are also sufficient. With budget multiplier $\Lambda \geq 0$ and nonnegativity multipliers $\tau_k \geq 0$, stationarity and complementary slackness give

$$w_k p_k'(F_k) - \Lambda\gamma_k + \tau_k = 0, \qquad \tau_k F_k = 0.$$

A funded coordinate thus satisfies $w_k p_k'(F_k) = \Lambda\gamma_k$, and an unfunded one satisfies $w_k p_k'(0) \leq \Lambda\gamma_k$. A differentiable strictly increasing concave function on $[0,\infty)$ has positive derivative at every finite argument: a zero derivative would preclude any subsequent increase by concavity. Since some coordinate is funded, $\Lambda > 0$.

For the shifted power law, $p_k'(F) = r d_k (1+F)^{-r-1}$. Set $g_k = (w_k d_k/\gamma_k)^{1/(1+r)}$ and $\mu = (r/\Lambda)^{1/(1+r)}$. Funded stationarity gives $1 + F_k = \mu g_k$, whereas exclusion gives $\mu g_k \leq 1$. Combining them proves

$$F_k^\star = [\mu g_k - 1]_+.$$

The budget function $\sum_k \gamma_k [\mu g_k - 1]_+$ is continuous, zero for sufficiently small μ, strictly increasing once positive, and unbounded above. Exactly one μ therefore makes the budget function equal to $B > 0$. □

For the funded set $\mathcal{K}_+ = \{k\colon \mu g_k > 1\}$,

$$\mu = \frac{B + \sum_{k\in\mathcal{K}_+} \gamma_k}{\sum_{k\in\mathcal{K}_+} \gamma_k g_k}. \tag{58}$$

The funded set is an upper level set of g_k, found by sorting and checking the corresponding sets. Equivalently, start with all coordinates, solve the displayed budget equation, and drop a smallest-g coordinate when $\mu g_k < 1$. Such a removal decreases μ: if $T = B + \sum \gamma_k$ and $U = \sum \gamma_k g_k$, then $(T - \gamma_j)/(U - \gamma_j g_j) < T/U$ if and only if $(T/U) g_j < 1$. A coordinate with equality has zero allocation and can be excluded without changing the solution.

J.3 Connecting a hard-budget subproblem to efficiency

Proposition 20 maximizes coverage value under a capacity constraint. It does not imply that an efficiency-optimal policy exhausts a computation budget. To connect the weights to J_λ, let family k have frequency ν_k, fixed hit/miss values, and fallback cost. Its benefit from increasing the hit probability has weight

$$w_k = \nu_k(v_{h,k} - v_{\mathrm{miss},k} + \lambda b_k).$$

Positive-margin families enter the concave coverage problem. If effort F_k also adds $\rho_k F_k$ to expected per-round resource, the appropriate objective, up to an additive constant, is

$$\sum_k w_k p_k(F_k) - \lambda \sum_k \rho_k F_k, \qquad \sum_k \gamma_k F_k \leq B.$$

The funded KKT condition becomes

$$w_k p_k'(F_k) = \lambda\rho_k + \Lambda\gamma_k,$$

with the corresponding weak inequality at zero. The budget need not bind in this penalized problem. The hard-budget result is recovered when allocation resource is separately constrained and has no additional variable charge in this objective, or when that charge is fixed across compared allocations.

Activation thresholds. The weight $w_k = \nu_k W_{\lambda,k}$ prices coverage only for a family that is routed cheap-first. By Proposition 8, family k is routed cheap-first at price λ if and only if $p_k(F_k)W_{\lambda,k} \geq \lambda a_k$; otherwise it dispatches directly and gains nothing from coverage. Relative to always dispatching directly, its benefit is therefore

$$\nu_k \max\{0,\ p_k(F_k)W_{\lambda,k} - \lambda a_k\}, \tag{59}$$

which is zero up to the activation threshold $F_k^{\min}$ defined by $p_k(F_k^{\min}) = \lambda a_k / W_{\lambda,k}$ (with $F_k^{\min} = \infty$ if this ratio exceeds one) and concave increasing beyond it. The kink at $F_k^{\min}$ makes (59) non-concave, so the Karush-Kuhn-Tucker conditions of Proposition 20 are not sufficient for the J_λ problem. Problem (56) is the inner problem for a fixed activation set $\mathcal{A}$: maximize $\sum_{k\in\mathcal{A}} w_k p_k(F_k)$ subject to $\sum_k \gamma_k F_k \leq B$, $F_k \geq F_k^{\min}$ for $k \in \mathcal{A}$ and $F_k = 0$ otherwise. For the shifted power law its solution is $F_k^\star = \max\{F_k^{\min}, \mu g_k - 1\}$ on $\mathcal{A}$, with μ set by the budget. The outer problem chooses $\mathcal{A}$ among the sets with $\sum_{k\in\mathcal{A}} \gamma_k F_k^{\min} \leq B$; at the resulting allocation the per-family routing decision is the state-wise rule of Theorem 2 with the family as the observable state, and only the coupling through the budget makes the outer choice combinatorial.

For a counterexample to allocating first and routing afterwards, take two equally frequent families with $\lambda = b_k = \gamma_k = 1$, $v_h = v_b$, $B = 2$, $p_k(F) = F/(1+F)$ and $a_1 = a_2 = 3/5$. The equal-weight solution of (56) is $F = (1, 1)$, but $p_k(1) = 1/2 < 3/5$ activates neither family, giving zero gain. Allocating $F = (2, 0)$ instead activates family 1 and gains $\frac{1}{2}(\frac{2}{3} - \frac{3}{5}) = 1/30 > 0$. The activation choice must therefore precede marginal equalization; pruning inactive families after the fact does not recover the optimum.

J.4 Static coverage and adaptive selection

For $S \subseteq T$ and $m \notin T$,

$$\begin{aligned} p(S \cup \{m\}) - p(S) &= \mathbb{P}\left(E_m \cap \bigcap_{i\in S} E_i^c\right) \\ &\geq \mathbb{P}\left(E_m \cap \bigcap_{i\in T} E_i^c\right) = p(T \cup \{m\}) - p(T). \end{aligned}$$

This proves diminishing returns under every fixed joint law. Nonnegative marginals also prove monotonicity. The classical static cardinality result consequently applies (Nemhauser et al., 1978); a price budget instead requires an appropriate submodular-knapsack method (Sviridenko, 2004).

Static submodularity does not establish adaptive submodularity. Let exactly one of two sources succeed, each with probability $1/2$. Before observations, the second source's marginal success probability is $1/2$. After the first source fails, the conditional marginal becomes one. Information has increased its marginal benefit, violating adaptive diminishing returns. An adaptive guarantee requires the additional property studied by Golovin & Krause (2011). Likewise, subtracting modular call cost preserves submodularity but need not preserve monotonicity, so the monotone-coverage approximation guarantees do not apply unchanged to the penalized objective.

K State-distribution Shifts

A floor fitted on probing tasks is an in-sample constraint. This section gives three complements to Theorem 2. Proposition 3 turns the fitted price path into a finite, ordered family of routers and certifies a floor on independent tasks. Proposition 21 asks which floors survive a change of task distribution. Corollary 22 combines the two: it certifies a dispatch floor that survives every shift of the state distribution, and our data show what that robustness costs.

K.1 Proof of Proposition 3

Setting. Let P be the deployment law of a task. A task has a state x in a finite set $\mathcal{X}$, a menu $\mathcal{U}(x)$, a realized value $V_u \in [0,1]$ and cost $D_u > 0$ for every $u \in \mathcal{U}(x)$, and the reference value $Y_B \in [0,1]$ and cost C_B. A router g, possibly randomized, is scored by its expected delivered value given the task, $V_g = \sum_u g(u \mid x) V_u$, which is $V_{g(x)}$ for a deterministic router, and $Q_g = \mathbb{E}_P[V_g - Y_B]$ is its quality change. The reference is admissible: it is a router g_B with $\mathbb{E}_P[V_{g_B}] = \mathbb{E}_P[Y_B]$, so $Q_{g_B} = 0$. Two samples are independent: $\mathcal{A}$ is arbitrary fitting data, and $\mathcal{B} = (T_1, \dots, T_n)$ consists of n i.i.d. tasks from P, each recording x_i, $Y_{B,i}$ and $V_{u,i}$ for every action used below, as in matched replay.

From $\mathcal{A}$ we fit $\hat{\pi}(x) > 0$, $\hat{v}(x,u) \in [0,1]$ and $\hat{c}(x,u) > 0$. For a router g write $\hat{\mathbb{E}}_g[V] = \sum_x \hat{\pi}(x) \sum_u g(u \mid x) \hat{v}(x,u)$, define $\hat{\mathbb{E}}_g[D]$ likewise, and put $\hat{\eta}(g) = \hat{\mathbb{E}}_g[V]/\hat{\mathbb{E}}_g[D]$ and $\hat{J}_\lambda(g) = \hat{\mathbb{E}}_g[V] - \lambda \hat{\mathbb{E}}_g[D]$. Let $\hat{\phi}_x(\lambda) = \max_{u \in \mathcal{U}(x)} [\hat{v}(x,u) - \lambda \hat{c}(x,u)]$ and $\hat{\Phi} = \sum_x \hat{\pi}(x) \hat{\phi}_x$. Theorem 2(i), applied to the fitted primitives, gives the root $\hat{\eta}^\star = \max_g \hat{\eta}(g)$, and we assume $\hat{\mathbb{E}}_{g_B}[V] > 0$, so that $\hat{\eta}^\star > 0$. A *breakpoint* of $\hat{\phi}_x$ is a point at which $\hat{\phi}_x$ is not differentiable. Let $\lambda_1 < \cdots < \lambda_{K-1}$ be the breakpoints of all $\hat{\phi}_x$ that lie in $(0, \hat{\eta}^\star)$, and set $\lambda_0 = 0$, $\lambda_K = \hat{\eta}^\star$ and $I_k = (\lambda_{k-1}, \lambda_k)$. Fix a strict order on each $\mathcal{U}(x)$ (the tie-break), and let $g_k(x)$ be the first maximizer of $\hat{v}(x,u) - \lambda \hat{c}(x,u)$ in that order, for $\lambda \in I_k$. The sequence $g_1, \dots, g_K$ is the *price path*. A mixture $w g_k + (1-w) g_{k+1}$ is the router that, in each state, plays $g_k(x)$ with probability w and $g_{k+1}(x)$ otherwise.

Proposition 3 (full statement). (a) Path. *(1) Each g_k is well defined, is the same for every $\lambda \in I_k$, and maximizes $\hat{J}_\lambda$ over all routers for every $\lambda \in [\lambda_{k-1}, \lambda_k]$. Moreover $K \le 1 + \sum_x (|\mathcal{U}(x)| - 1)$. (2) For $k < l$, $\hat{\mathbb{E}}_{g_k}[V] > \hat{\mathbb{E}}_{g_l}[V]$, $\hat{\mathbb{E}}_{g_k}[D] > \hat{\mathbb{E}}_{g_l}[D]$ and $\hat{\eta}(g_k) < \hat{\eta}(g_l)$; in particular $g_1, \dots, g_K$ are distinct. (3) For every $\epsilon \ge 0$, the fitted floor problem $\max\{\hat{\eta}(g) : \hat{\mathbb{E}}_g[V] \ge \hat{\mathbb{E}}_{g_B}[V] - \epsilon\}$ of Theorem 2(iii) has an optimal router that is some g_k or a mixture of g_k and g_{k+1}.*

(b) Certification. *Let $\alpha \in (0,1)$ and $\epsilon \ge 0$, and let $p_1, \dots, p_K$ be functions of $(\mathcal{A}, \mathcal{B})$ such that $\mathbb{P}(p_k \le \alpha \mid \mathcal{A}) \le \alpha$ whenever $Q_{g_k} < -\epsilon$. Let $\hat{k} = \max\{k \in \{0, \dots, K\} : p_j \le \alpha \text{ for all } j \le k\}$, and let $\hat{g} = g_{\hat{k}}$ if $\hat{k} \ge 1$ and $\hat{g} = g_B$ otherwise. Then*

$$\mathbb{P}\big(Q_{\hat{g}} < -\epsilon \mid \mathcal{A}\big) \le \alpha \quad \textit{almost surely, hence} \quad \mathbb{P}\big(Q_{\hat{g}} < -\epsilon\big) \le \alpha.$$

With $\hat{Q}_k = n^{-1} \sum_{i=1}^n (V_{g_k,i} - Y_{B,i})$, the Hoeffding p-value $p_k = \exp\{-n(\hat{Q}_k + \epsilon)_+^2/2\}$ satisfies the requirement. The conclusion holds for any finite family of routers and any order that are fixed given $\mathcal{A}$.

Proof. Part (a)(1). Fix x. The function $\hat{\phi}_x$ is the maximum of $|\mathcal{U}(x)|$ affine functions with slopes $-\hat{c}(x,u) < 0$. It is therefore convex and piecewise linear, and each one-sided derivative $\hat{\phi}'_x(\lambda\pm)$ equals the slope of an affine function that is active at λ. By convexity, $\hat{\phi}'_x(\lambda-) \le \hat{\phi}'_x(\lambda+)$, with strict inequality exactly at a breakpoint. Let $b_1 < \cdots < b_m$ be the breakpoints of $\hat{\phi}_x$. On each open interval (b_j, b_{j+1}) the function has no breakpoint, so it is affine there and $\hat{\phi}'_x(b_j+) = \hat{\phi}'_x(b_{j+1}-)$. Hence

$$\hat{\phi}'_x(b_1-) < \hat{\phi}'_x(b_1+) = \hat{\phi}'_x(b_2-) < \hat{\phi}'_x(b_2+) = \cdots < \hat{\phi}'_x(b_m+),$$

so these are $m+1$ distinct elements of $\{-\hat{c}(x,u) : u \in \mathcal{U}(x)\}$. Hence $m \le |\mathcal{U}(x)| - 1$. A breakpoint shared by several states is counted once, so $K - 1 \le \sum_x (|\mathcal{U}(x)| - 1)$.

On I_k no $\hat{\phi}_x$ has a breakpoint, so $\hat{\phi}_x(\lambda) = a_x - b_x \lambda$ on I_k. For each u, the function $\delta_u(\lambda) = \hat{\phi}_x(\lambda) - [\hat{v}(x,u) - \lambda \hat{c}(x,u)]$ is affine and nonnegative on I_k. If it vanishes at one point of the open interval I_k, it vanishes on all of I_k. So the maximizer set $M_x(\lambda) = \arg\max_u [\hat{v}(x,u) - \lambda \hat{c}(x,u)]$ is the same for every $\lambda \in I_k$, and so is its first element $g_k(x)$. Every $u \in M_x(\lambda)$, $\lambda \in I_k$, then satisfies $(\hat{v}(x,u), \hat{c}(x,u)) = (a_x, b_x)$. By fact (F1) of Appendix H.3, g_k maximizes $\hat{J}_\lambda$ for $\lambda \in I_k$. Both $\hat{J}_\lambda(g_k)$ and $\hat{\Phi}(\lambda)$ are continuous in λ and agree on I_k, so they agree on its closure.

Part (a)(2). Fix $1 \le k \le K-1$. Let $\mathcal{X}_k$ be the set of states for which λ_k is a breakpoint; it is nonempty by the definition of λ_k. For $x \notin \mathcal{X}_k$, $\hat{\phi}_x$ is affine on $I_k \cup \{\lambda_k\} \cup I_{k+1}$, so by (a)(1) the actions $g_k(x)$ and $g_{k+1}(x)$ have the same pair $(\hat{v}, \hat{c})$. For $x \in \mathcal{X}_k$, $\hat{c}(x, g_k(x)) = -\hat{\phi}'_x(\lambda_k-) > -\hat{\phi}'_x(\lambda_k+) =$

$\hat{c}(x, g_{k+1}(x))$, and both actions attain $\hat{\phi}_x(\lambda_k)$ by continuity, so $\hat{v}(x, g_{k+1}(x)) - \hat{v}(x, g_k(x)) = \lambda_k[\hat{c}(x, g_{k+1}(x)) - \hat{c}(x, g_k(x))]$. Write $(v_j, d_j) = (\hat{\mathbb{E}}_{g_j}[V], \hat{\mathbb{E}}_{g_j}[D])$. Because $\hat{\pi} > 0$ and $\lambda_k > 0$,

$$d_{k+1} - d_k = \sum_{x \in \mathcal{X}_k} \hat{\pi}(x)\big[\hat{c}(x, g_{k+1}(x)) - \hat{c}(x, g_k(x))\big] < 0, \qquad v_{k+1} - v_k = \lambda_k(d_{k+1} - d_k) < 0.$$

Chaining consecutive steps gives $v_k > v_l$ and $d_k > d_l$ for $k < l$. For efficiency, pick $\lambda' \in I_l$. Optimality of g_l at λ' gives $v_k - v_l \le \lambda'(d_k - d_l)$, hence

$$v_l d_k - v_k d_l = v_l(d_k - d_l) - d_l(v_k - v_l) \ge (d_k - d_l)(v_l - \lambda' d_l) = (d_k - d_l)\,\hat{\Phi}(\lambda') > 0,$$

because $d_k > d_l$ and $\hat{\Phi}$ is strictly decreasing with root $\hat{\eta}^\star > \lambda'$ (Theorem 2(i)). Dividing by $d_k d_l > 0$ gives $\hat{\eta}(g_l) > \hat{\eta}(g_k)$. Distinct fitted costs make the routers distinct.

Part (a)(3). Apply Theorem 2(iii) to the fitted primitives. It gives a multiplier $\kappa \ge 0$, and every constrained optimum uses only actions that maximize $(1+\kappa)\hat{v} - \hat{\eta}^\star_\epsilon \hat{c}$. Equivalently, by (F1), every constrained optimum maximizes $\hat{J}_{\lambda_\epsilon}$ with $\lambda_\epsilon = \hat{\eta}^\star_\epsilon/(1+\kappa)$. Here $\lambda_\epsilon \ge \hat{\eta}(g_B)/(1+\kappa) > 0$ and $\lambda_\epsilon \le \hat{\eta}^\star_\epsilon \le \hat{\eta}^\star$. Two facts about ties are used. Let $u, u' \in M_x(\mu)$ for some $\mu > 0$. If u' also maximizes on an interval just to the left of μ, then $\hat{c}(x, u') \ge \hat{c}(x, u)$: the affine difference $[\hat{v}(x, u') - \lambda \hat{c}(x, u')] - [\hat{v}(x, u) - \lambda \hat{c}(x, u)]$ is nonnegative there and zero at μ, so its slope is nonpositive. Tying at μ then gives $\hat{v}(x, u') - \hat{v}(x, u) = \mu[\hat{c}(x, u') - \hat{c}(x, u)] \ge 0$. If u' maximizes just to the right of μ, both inequalities reverse.

Case $\kappa = 0$. By the proof of Theorem 2(iii), $\kappa = 0$ is a valid multiplier only if some efficiency-optimal router meets the floor, and then $\hat{\eta}^\star_\epsilon = \hat{\eta}^\star$; so some efficiency-optimal router satisfies the floor. The router g_K maximizes $\hat{J}_{\hat{\eta}^\star}$ by (a)(1), so it is efficiency-optimal by Theorem 2(i). In each state it maximizes on I_K, just to the left of $\hat{\eta}^\star$, so by the tie facts it has the largest fitted value among the actions tied at $\hat{\eta}^\star$. Hence $\hat{\mathbb{E}}_{g_K}[V] \ge \hat{\mathbb{E}}_h[V]$ for every maximizer h of $\hat{J}_{\hat{\eta}^\star}$. Because some such h satisfies the floor, g_K does too, and g_K is optimal.

Case $\kappa > 0$. Then $0 < \lambda_\epsilon < \hat{\eta}^\star$. Let g^* be a constrained optimum. If $\lambda_\epsilon \in I_k$, every action in $M_x(\lambda_\epsilon)$ has the pair $(\hat{v}, \hat{c})$ of $g_k(x)$, by (a)(1). So g^* and g_k have the same fitted value and cost, and g_k is optimal. Otherwise $\lambda_\epsilon = \lambda_k$ with $1 \le k \le K-1$. By continuity, $g_k(x)$ and $g_{k+1}(x)$ both lie in $M_x(\lambda_k)$. By the tie facts, $\hat{c}(x, g_{k+1}(x)) \le \hat{c}(x, u) \le \hat{c}(x, g_k(x))$ for every $u \in M_x(\lambda_k)$, so $d_{k+1} \le \hat{\mathbb{E}}_{g^*}[D] \le d_k$. Every maximizer h of $\hat{J}_{\lambda_k}$ satisfies $\hat{\mathbb{E}}_h[V] = \hat{\Phi}(\lambda_k) + \lambda_k \hat{\mathbb{E}}_h[D]$. Choose $w \in [0,1]$ with $\hat{\mathbb{E}}_{g^*}[D] = w d_k + (1-w) d_{k+1}$. Fitted value and cost are linear in the router, so the mixture $w g_k + (1-w) g_{k+1}$ has the fitted value and cost of g^*. It therefore has the same efficiency, satisfies the floor, and is optimal.

Part (b). Condition on $\mathcal{A}$. Then $g_1, \dots, g_K$, their order, and the numbers Q_{g_k} are fixed. Let $k_0 = \min\{k : Q_{g_k} < -\epsilon\}$, with $k_0 = \infty$ if no path router violates the floor. The reference cannot violate it, since $Q_{g_B} = 0 \ge -\epsilon$; this is the only use of $\epsilon \ge 0$. Suppose $Q_{\hat{g}} < -\epsilon$. Then $\hat{k} \ge 1$ and $g_{\hat{k}}$ violates the floor, so $k_0 \le \hat{k} < \infty$. By the definition of $\hat{k}$, every p_j with $j \le \hat{k}$ is at most α, and in particular $p_{k_0} \le \alpha$. Therefore

$$\mathbb{P}\big(Q_{\hat{g}} < -\epsilon \mid \mathcal{A}\big) \le \mathbf{1}\{k_0 < \infty\}\,\mathbb{P}\big(p_{k_0} \le \alpha \mid \mathcal{A}\big) \le \alpha,$$

by the requirement on p_{k_0}, because $Q_{g_{k_0}} < -\epsilon$. Taking expectations over $\mathcal{A}$ gives the unconditional bound. The argument uses only that the family is finite and that it and its order are fixed given $\mathcal{A}$. It places no condition on the dependence among $p_1, \dots, p_K$, which share the sample $\mathcal{B}$.

For the Hoeffding p-value, condition on $\mathcal{A}$. Since $\mathcal{B}$ is independent of $\mathcal{A}$ and g_k is fixed, the variables $Z_i = V_{g_k,i} - Y_{B,i}$ are i.i.d., take values in $[-1, 1]$, and have mean Q_{g_k}. Put $t_\alpha = \sqrt{2\log(1/\alpha)/n} > 0$. Then $p_k \le \alpha$ if and only if $\hat{Q}_k + \epsilon \ge t_\alpha$. If $Q_{g_k} < -\epsilon$, Hoeffding's inequality for a range of length two (Hoeffding, 1963) gives

$$\mathbb{P}(p_k \le \alpha \mid \mathcal{A}) \le \mathbb{P}\big(\hat{Q}_k - Q_{g_k} \ge t_\alpha \mid \mathcal{A}\big) \le \exp(-n t_\alpha^2/2) = \alpha. \qquad \square$$

Order, power, and other p-values. Validity does not use the monotonicity in part (a). The order matters for power, because the search stops at the first $p_k > \alpha$. Part (a)(2) puts routers with high fitted value, and therefore a small risk of violation, first. For the Hoeffding test, the cost of certification is a

margin t_α. If $Q_{g_j} \geq -\epsilon + t_\alpha + \delta$ for all $j \leq m$, then $p_j > \alpha$ requires $\hat{Q}_j - Q_{g_j} < -\delta$. A union bound gives $\mathbb{P}(\hat{k} \geq m \mid \mathcal{A}) \geq 1 - m\, e^{-n\delta^2/2}$. At $n = 3{,}540$ and $\alpha = 0.05$, $t_\alpha = 0.041$. Any other valid p-value can be used. The empirical-Bernstein bound of Maurer & Pontil (2009, Theorem 4), stated for i.i.d. variables in $[0, 1]$ and applied to $(1 - Z_i)/2$, yields a valid p-value whose margin shrinks with the sample variance of $V - Y_B$. A normal approximation is valid only asymptotically.

Refining the path. Because validity needs only a finite family in an order fixed by $\mathcal{A}$, mixtures $wg_k + (1 - w)g_{k+1}$ with decreasing w may be inserted between consecutive path routers. We insert the grid $w \in \{0.9, 0.8, \ldots, 0.1\}$ and, for each tolerance ϵ studied, the exact weight $w^*(\epsilon)$ of the fitted floor optimum of part (a)(3), computed on $\mathcal{A}$. Fitted value and cost remain monotone along the refined family, and by construction the family contains a fitted floor optimum at every tolerance studied. A mixture is scored by its expected per-task value, $Z_i = wZ_{k,i} + (1 - w)Z_{k+1,i} \in [-1, 1]$, whose mean is the mixture's quality change. The length of the family affects only how long the search runs, not its level.

Certified price and data-chosen margin. If $\hat{k} \geq 1$, the certified router maximizes $\hat{J}_\lambda$ at $\hat{\lambda} = \lambda_{\hat{k}}$, the right end of its interval on the path; a mixture of g_k and g_{k+1} maximizes it only at λ_k. For $\hat{k} < K$ this is the largest price at which $g_{\hat{k}}$ maximizes $\hat{J}_\lambda$, because in the states of $\mathcal{X}_{\hat{k}}$ it is strictly worse than $g_{\hat{k}+1}$ above $\lambda_{\hat{k}}$; the last router g_K may remain a maximizer above $\hat{\eta}^\star$. No price is returned with the reference. We call $\hat{\lambda}$ the *certified price*. It turns the choice of λ into an output of the reference, the tolerance ϵ, and the level α, which is the role Theorem 2(iii) gives to λ_ϵ in sample. The certified router's fitted margin over the fitted floor is $\hat{m} = \hat{\mathbb{E}}_{\hat{g}}[V] - (\hat{\mathbb{E}}_{g_B}[V] - \epsilon)$. Nothing forces $\hat{m} \geq 0$: when the estimates on $\mathcal{B}$ exceed those on $\mathcal{A}$, the search can certify past the last router that meets the fitted floor. In our splits, $\hat{m}$ is positive in at least 90% of splits for every finite-sample certified router (10th percentile 0.013 for the refined Bernstein router at $\epsilon = 0$), but this is an observation, not a guarantee. Certification can therefore be read as a margin rule in which the held-out test, not a resampling search, chooses the margin, and in which the certified router, not the margin, comes with a finite-sample guarantee.

What is and is not guaranteed. The guarantee concerns the population quality change $Q_{\hat{g}}$ under the law P that generated $\mathcal{B}$. It holds over the joint draw of $\mathcal{A}$ and $\mathcal{B}$, and it requires i.i.d. tasks with matched outcomes on $\mathcal{B}$ for every action the path uses. It does not cover a different deployment law, which Proposition 21 and Corollary 22 address. It needs $\epsilon \geq 0$ because the fallback must meet the floor. For $\epsilon < 0$, which would certify a strict gain, the reference violates the floor; replacing the fallback by abstention gives $\mathbb{P}(\hat{k} \geq 1,\ Q_{g_{\hat{k}}} < -\epsilon) \leq \alpha$ by the same proof. It also says nothing about efficiency beyond part (a)(2): $\hat{k}$ selects the most fitted-efficient certified router in the family.

Numerical illustration on CodeRouterBench. We use the 21-cell, 22-action menu of Section 3.1. Fitted on all 7,080 probing tasks, the price path has $K = 38$ routers, against the bound $1 + 21 \cdot 21 = 442$, and each step changes the action in exactly one cell. Along the path, fitted value and cost strictly decrease and fitted efficiency strictly increases from 2.60 to 29.42 times the fitted reference efficiency, as part (a)(2) requires. The zero-tolerance floor price $\lambda_\epsilon = 144.3 = 4.16\,\eta_B$ of Appendix A.4 ($\kappa = 5.43$) is exactly the breakpoint between g_{35}, with fitted $\Delta V = +0.028$, and g_{36}, with -0.004. The fitted floor optimum mixes them with weight 0.12 on g_{35} and has fitted efficiency $26.71\,\eta_B$, the value of the Charnes–Cooper linear program, as the case $\kappa > 0$ of part (a)(3) predicts.

For certification, we draw an i.i.d. random half of probing as the fitting sample $\mathcal{A}$ (3,540 tasks) and use the other half as $\mathcal{B}$ (3,540 tasks), without stratifying, so that $\mathcal{B}$ is i.i.d. from the probing law as part (b) assumes. We fit the path on $\mathcal{A}$, test on $\mathcal{B}$ at $\alpha = 0.05$, and evaluate the selected router on the test set with 1,000 paired task bootstraps (seed 20260925). We repeat this over 200 random splits. Certification needs the outcome of every path action and of the reference on $\mathcal{B}$; running the models the path uses, Opus included, costs \$0.0158 per task, or \$55.86 for the 3,540 tasks of $\mathcal{B}$ at token prices, against \$43.81 for Opus alone.

Two uncertified baselines use the same fitted path. The Theorem 2(iii) floor optimum refitted on $\mathcal{A}$, the mixture of part (a)(3), keeps test $\Delta V \geq -\epsilon$ in only 12.5%, 15.0% and 15.5% of splits at $\epsilon = 0$, 0.005 and 0.01, with median η/η_B of 25.63 (10th–90th percentile 23.83–27.10), 27.05 and 27.60. Its deterministic round-down, the last path router that meets the fitted floor, keeps it in 76.5%, 35.5%

and 63.0% of splits, with median 19.34 (18.76–26.85) at $\epsilon = 0$. The round-down lands past the large efficiency jump, at a ratio above $25\times$, in 23.5% of splits.

With certification, the selected router keeps $\Delta V \geq -\epsilon$ on the test set in all 200 splits for every ϵ and every finite-sample p-value. The medians of η/η_B across splits at $\epsilon = 0$ are:

- Hoeffding: 15.17 (15.12–15.22);
- empirical Bernstein: 18.83 (15.17–19.41);
- empirical Bernstein on the mixture-refined path: 19.38 (18.33–20.63).

On the refined path, $\epsilon = 0.005$ and 0.01 raise the median to 20.47 (19.33–21.78) and 21.54 (20.42–23.00). The refined Bernstein router has median certified price $\hat{\lambda} = 4.21\,\eta_B$ (3.99–4.43) and median fitted margin $\hat{m} = 0.026$ (0.013–0.037) at $\epsilon = 0$. It thus prices just below the uncertified floor optimum, whose median price refitted on $\mathcal{A}$ is $4.34\,\eta_B$. Near this price efficiency is very sensitive: on full probing, the one-cell step from g_{35} to g_{36} raises fitted efficiency from 20.12 to $28.05\,\eta_B$. Its margin is close to the 0.0275–0.03 that a one-sided 95% probing lower bound selects in Appendix A.4. Bonferroni selection over the same path, which is also valid, gives medians of 13.29 with Hoeffding and 15.17 with Bernstein, so the fixed order adds 1.88 and 3.66 to the median η/η_B. On the unrefined path, the Bernstein router coincides with the round-down in 69% of splits. Certification changes the outcome mainly in the splits in which the round-down overshoots. The asymptotic normal-approximation test keeps $\Delta V \geq -\epsilon$ in 96.5–100% of splits.

The seed-20260925 split is one in which both uncertified baselines overshoot. The Theorem 2(iii) optimum gives $27.68\times$ $[26.32, 29.06]$ with $\Delta V = -0.0128$ $[-0.0263, -0.0001]$ and bootstrap compliance 0.025, and the round-down gives $27.33\times$ with -0.0089 $[-0.0221, +0.0039]$ and compliance 0.069. The refined Bernstein router at $\epsilon = 0$ gives $17.91\times$ $[16.90, 19.04]$ at 0.675 m$ per task, with $\Delta V = +0.0161$ $[+0.0032, +0.0290]$, compliance 0.986 and $\hat{\lambda} = 4.23\,\eta_B$. At $\epsilon = 0.01$ it gives $19.37\times$ $[18.29, 20.56]$ at 0.613 m$, with $\Delta V = +0.0082$ $[-0.0052, +0.0213]$ and compliance 0.997. The Hoeffding router at $\epsilon = 0$ gives $15.19\times$ $[14.29, 16.16]$ with $\Delta V = +0.0414$ $[+0.0286, +0.0540]$. The certified routers therefore sit near the probing-margin rule of Section 3.1, $18.46\times$ at margin 0.03 (compliance 0.965 with this bootstrap seed). They fit on half of probing, but their floor has a finite-sample guarantee and needs no margin search.

Test compliance measures the uncertainty of the test sample, not the population violation that part (b) bounds. Moreover, the guarantee is for the probing law, and the test set's cell shares differ from probing's by a total-variation distance of 0.032, a small state shift in the sense of Proposition 21(i). The 200-split pattern is therefore consistent with the guarantee but is not a test of its level.

Nested resplits. We also pool the 9,999 in-distribution tasks and draw 200 outer 70/30 splits stratified by the 21 cells. In each split, all policy fitting uses the outer fitting partition, with an unstratified inner half-split for path fitting and certification. At $\epsilon = 0$, the refined empirical-Bernstein procedure attains test $\Delta V \geq 0$ in 199/200 outer splits, with median $\eta/\eta_B = 20.56$ (2.5th–97.5th percentile 18.25–22.37) and median $\Delta V = +0.0215$. Refitting the wrapper-stripping rule, TF–IDF vocabulary and cell classifier within each outer fitting partition gives predicted-cell routing a median $20.16\times$ (17.31–21.87), median $\Delta V = +0.0213$, and the same 199/200 attainment. These are empirical frequencies and resplit ranges from overlapping partitions of one task pool; they describe stability across partitions rather than a new population guarantee.

Learned router paths. The same empirical-Bernstein fixed sequence can select along a learned router's increasing price grid. We evaluate this workflow at $\epsilon = 0$ and $\alpha = 0.05$ on stratified half-splits of probing, then on the fixed 2,919-task test set. A bge-large multi-head model is retrained and its predictions calibrated on $\mathcal{A}$ in each split; the path is tested on $\mathcal{B}$, and the selected $\mathcal{A}$-trained router is deployed. Direct dispatch attains test $\Delta V \geq 0$ in all 60 evaluated splits, with median $17.48\times$ efficiency (10th–90th percentile 14.13–19.54) and median $\Delta V = +0.0201$; the full action menu gives $16.27\times$ and 60/60 attainment. A frozen-bge ridge model refitted and calibrated on $\mathcal{A}$ gives 200/200 attainment, with median efficiencies $10.03\times$ for direct dispatch and $11.96\times$ for the full menu. Its penalties are retained from full-probing cross-validation. Both models use fixed wrapper-removal rules learned from probing prompts. These replay results extend the empirical evidence to task-level learned predictions; the attainment counts retain the test-sample interpretation above.

K.2 WHICH QUALITY FLOORS TRANSFER

Setting. A target law P' describes a task: its state x in a finite set $\mathcal{X}$, the values and costs of the available actions, verifier outputs, and the reference value Y_B and cost C_B. All values lie in $[0,1]$. Primes denote quantities under P': $Q' = \mathbb{E}'[V - Y_B]$ and $\mathcal{S}' = \mathbb{E}'[C_B - D]$. An *ex-ante router* g plays action $u \in \mathcal{U}(x)$ with probability $g(u \mid x)$, independently of the outcomes given x, and delivers that action's value V_u. It is an *ex-ante dispatcher* if every action is a single model call whose output is delivered unchecked, so that $V_u = Y_u$. The reference is a single action B. Whenever B is played, or called as a fallback, it delivers Y_B for every realization, under the source law and every target law (matched outcomes); so $V_B = Y_B$ pointwise. We set $g(B \mid x) = 0$ when $B \notin \mathcal{U}(x)$. For a source law P, write $\Delta(x,u) = \mathbb{E}_P[V_u - Y_B \mid x]$ and $q(x) = \sum_u g(u \mid x)\Delta(x,u)$.

A *reference-terminated cascade* runs stages $k = 1, \dots, K$ in order. Stage k calls a primary with value Y_k, and an acceptance rule decides whether to stop with that output. The rule may depend on verifier outputs, the history, or external randomness. Let $H_k = 1$ if stage k is reached and accepted. If no stage accepts, the reference is called and delivers Y_B, the value the reference policy would deliver on the same task. The one-stage case is Eq. (5), with $H = H_1$ and $Y_P = Y_1$.

Proposition 21 (Which quality floors transfer)**.** (i) Ex-ante routing. *Let $\mathcal{P}_{\mathrm{state}}(P)$ be the set of target laws whose outcome law given x equals P's and whose state law π' is arbitrary. For any ex-ante router, $\inf_{P' \in \mathcal{P}_{\mathrm{state}}(P)} Q' = \min_x q(x)$, attained at a point mass. Consequently, $Q' \geq -\epsilon$ for every $P' \in \mathcal{P}_{\mathrm{state}}(P)$ if and only if $q(x) \geq -\epsilon$ for every x. If outcome laws may also change and g is an ex-ante dispatcher, then for every state law π' the minimum of Q' over target laws with state law π' equals $-\sum_x \pi'(x)\,(1 - g(B \mid x))$. It is attained when $Y_B = 1$ and $Y_u = 0$ for every $u \neq B$. An action that delivers a primary only when the primary's own score is at least θ, and the reference otherwise, loses at most $1 - \theta$ per task under every law, so for $\theta = 1$ it is reference-sound.*

(ii) Reference-terminated cascades. *Pointwise, $V - Y_B = \sum_k H_k(Y_k - Y_B)$. Hence, for every target law P',*

$$Q' = \mathbb{E}'\Big[\sum_k H_k(Y_k - Y_B)_+\Big] - L', \qquad L' = \mathbb{E}'\Big[\sum_k H_k(Y_B - Y_k)_+\Big],$$

so $Q' \geq -L'$. If $P'(H_k = 1,\, Y_k < Y_B) = 0$ for every k, then $Q' \geq 0$. In particular, if acceptance is reference-sound, meaning that $H_k = 1$ implies $Y_k \geq Y_B$ for every realization, then $Q' \geq 0$ for every P'. For binary values and one stage,

$$L' = \mathrm{FAR}^{\mathrm{d}}_{P'}\, P'(Y_P = 0, Y_B = 1) \leq \mathrm{FPR}_{P'}\, P'(Y_P = 0),$$

where $\mathrm{FAR}^{\mathrm{d}}_{P'} = P'(H = 1 \mid Y_P = 0, Y_B = 1)$ is the false-accept rate on discordant tasks and $\mathrm{FPR}_{P'} = P'(H = 1 \mid Y_P = 0)$. If the law of H given (Y_P, Y_B) is the same under P' as under P, then $L' = \mathrm{FAR}^{\mathrm{d}}_{P}\, P'(Y_P = 0, Y_B = 1)$.

(iii) Saving. *For one stage, $\mathcal{S}' = \mathbb{E}'[HC_B] - \mathbb{E}'[A] = P'(H = 1)\,\mathbb{E}'[C_B \mid H = 1] - \mathbb{E}'[A]$. No acceptance rule guarantees $\mathcal{S}' \geq 0$ on every target. On any target on which the rule rejects everything and $\mathbb{E}'[A], \mathbb{E}'[Y_B] > 0$, $Q' = 0$, $\mathcal{S}' = -\mathbb{E}'[A] < 0$ and $\eta' < \eta'_B$. On any target on which every output is accepted and $A < C_B$, $\mathcal{S}' > 0$. If acceptance is reference-sound and $\mathbb{E}'[HC_B] \geq \mathbb{E}'[A]$ (equivalently $P'(H = 1) \geq \mathbb{E}'[A]/\mathbb{E}'[C_B \mid H = 1]$ when $P'(H = 1) > 0$), then $\mathcal{S}' \geq 0$ and $\eta' \geq \eta'_B$. With constant costs $A = a$ and $C_B = b$, the last condition reads $P'(H = 1) \geq a/b$. In general, $\eta' \geq \eta'_B$ if and only if $Q' + \eta'_B \mathcal{S}' \geq 0$, so the break-even condition is sufficient under reference-sound acceptance, and $\mathcal{S}' \geq 0$ is necessary when $Q' = 0$; a positive Q' can offset a negative saving.*

Proof. Part (i). For $P' \in \mathcal{P}_{\mathrm{state}}(P)$ with state law π', the action is independent of the outcomes given x, and the outcome law given x is P's. So $\mathbb{E}'[\mathbf{1}\{u \text{ played}\}(V_u - Y_B) \mid x] = g(u \mid x)\Delta(x,u)$ and $Q' = \sum_x \pi'(x) q(x)$. This is a convex combination of the numbers $q(x)$. It is at least $\min_x q(x)$, with equality when π' is the point mass at a minimizing state. Both claims about $\mathcal{P}_{\mathrm{state}}(P)$ follow; they use only that the action is chosen from x alone, so they hold for any menu, including cascade actions.

For the claim about outcome shifts, fix π' and let g be an ex-ante dispatcher. Pointwise, $V - Y_B = \sum_{u \neq B} \mathbf{1}\{u \text{ called}\}(Y_u - Y_B)$, since the term for $u = B$ vanishes by the matched-outcome coupling $V_B = Y_B$. (Without that coupling a target law could give the router's call to B value 0 while $Y_B = 1$,

and the minimum would be -1 for every g.) Each summand is at least $-\mathbf{1}\{u \text{ called}\}$ because $Y_u \geq 0$ and $Y_B \leq 1$. Taking expectations gives $Q' \geq -\sum_x \pi'(x)(1 - g(B \mid x))$. The outcome law with $Y_B = 1$ and $Y_u = 0$ for $u \neq B$ is admissible in every state and attains the bound. For a threshold action, $V - Y_B = H(Y_P - Y_B)$ with $H = \mathbf{1}\{Y_P \geq \theta\}$, and $H(Y_P - Y_B) \geq -H(1-\theta) \geq -(1-\theta)$.

Part (ii). At most one stage accepts. If one does, the delivered value is its Y_k; otherwise it is Y_B. So $V = \sum_k H_k Y_k + (1 - \sum_k H_k) Y_B$, which gives the pointwise identity. Write $y = y_+ - (-y)_+$ in each summand and take expectations under P'; every term is bounded. Then Q' is a nonnegative term minus L'. If $P'(H_k = 1, Y_k < Y_B) = 0$ for every k, then $L' = 0$. Pointwise soundness gives this under every P'. For binary values and one stage, $H(Y_B - Y_P)_+ = H\,\mathbf{1}\{Y_P = 0, Y_B = 1\}$, whose expectation is $\mathrm{FAR}^{\mathrm{d}}_{P'} P'(Y_P = 0, Y_B = 1)$, and $H\,\mathbf{1}\{Y_P = 0, Y_B = 1\} \leq H\,\mathbf{1}\{Y_P = 0\}$, whose expectation is $\mathrm{FPR}_{P'} P'(Y_P = 0)$. Under the invariance assumption, $\mathrm{FAR}^{\mathrm{d}}_{P'} = \mathrm{FAR}^{\mathrm{d}}_{P}$.

Part (iii). From Eq. (5), $C_B - D = HC_B - A$, and $\mathbb{E}'[HC_B] = P'(H = 1)\mathbb{E}'[C_B \mid H = 1]$. If $H = 0$ almost surely under P', then $V = Y_B$, so $Q' = 0$ and $\mathcal{S}' = -\mathbb{E}'[A] < 0$. Since $\mathbb{E}'[Y_B] > 0$, Lemma 1 then gives $\eta' - \eta'_B = \eta'_B \mathcal{S}'/\mathbb{E}'[D] < 0$. If $H = 1$ and $A < C_B$ almost surely, then $\mathcal{S}' = \mathbb{E}'[C_B - A] > 0$. Under reference-sound acceptance, $Q' \geq 0$ by part (ii). If also $\mathcal{S}' \geq 0$, Lemma 1 gives $\eta' \geq \eta'_B$. For the general statement, Lemma 1 under P' (with $\mathbb{E}'[Y_B] > 0$, so $\eta'_B > 0$) gives $\eta' - \eta'_B = (Q' + \eta'_B \mathcal{S}')/\mathbb{E}'[D]$, whose sign is that of $Q' + \eta'_B \mathcal{S}'$; when $Q' = 0$ it is the sign of $\mathcal{S}'$. The saving need not be nonnegative for efficiency to transfer. Take binary values, constant costs $a = 0.1$ and $b = 1$, and a reference-sound rule on a target with three task types: with probability 0.05, $Y_P = 1$, $Y_B = 0$ and $H = 1$; with probability 0.50, $Y_P = 0$, $Y_B = 1$ and $H = 0$; with probability 0.45, $Y_P = Y_B = 0$ and $H = 0$. Then $P'(H = 1) = 0.05 < a/b$ and $\mathcal{S}' = -0.05$, yet $\mathbb{E}'[V] = 0.55$, $\mathbb{E}'[D] = 1.05$ and $\eta' = 0.524 > \eta'_B = 0.5$. For K stages, the same computation gives $\mathcal{S}' = \mathbb{E}'[(\sum_k H_k) C_B] - \mathbb{E}'[A]$, where A is the total cost of the stages that are reached. □

Numerical illustration. *Part (i).* Fitted on all probing tasks, the path router g_{35}, the deterministic higher-value neighbour of the zero-tolerance floor optimum, has fitted $\Delta V = +0.028$ and test $\Delta V = +0.010$. It dispatches every cell directly to a non-reference model, so under outcome shifts its worst case is $Q' = -1$. Its cell-wise quality change $q(x)$ is negative in 8 of 21 cells, which hold 49.1% of probing tasks. The worst cell, code refactoring on Bugs2Fix, has -0.233 on probing and -0.241 on the test set. A shift of the state distribution alone toward that cell would therefore cost up to 0.23 score, although the aggregate floor holds.

Part (ii). All values in the 334-task verifier sample are binary. Take Qwen3-Max with a judge at $\theta = 0.5$. Its $Q = -0.0120$ $[-0.0419, +0.0210]$ is a gain of 0.0329 minus a false-accept loss $L = 0.0449$ $[0.0240, 0.0689]$. The FPR bound is 0.192, loose because Opus also fails on many primary failures. The discordant form is tighter. On the 69 discordant tasks of the 500-task probing verifier sample, the judge accepts 46.4% (one-sided 95% Clopper–Pearson upper bound 0.569). Multiplying that upper bound by the test set's discordance rate of 0.108 gives 0.061 under the invariance assumption, and the realized L is 0.045. This plug-in bound holds for six of the eight deployable verifier–primary pairs at $\theta = 0.5$. It fails for hidden tests with Kimi-K2.5 ($L = 0.042$ against 0.032) and Qwen3.5-Plus (0.030 against 0.028), whose discordant false-accept rates rise from 0.079 and 0.162 on probing to 0.179 and 0.312 on the test set. Invariance is thus an assumption about the target, not a property of the verifier. Visible tests at $\theta = 0.5$ give $Q = -0.0359$ $[-0.0659, -0.0030]$ with $L = 0.0629$ $[0.0359, 0.0868]$, and the Kimi-K2.5 judge gives $Q = -0.0389$ $[-0.0719, -0.0090]$ with $L = 0.0629$. Released-score acceptance is reference-sound by construction, with $L = 0$ and $Q = +0.0449$ $[+0.0240, +0.0689]$ for Qwen3-Max. Re-executing the grader's hidden tests at $\theta = 1$ also gives $L = 0$ for Qwen3-Max on this sample, an empirical count rather than a pointwise property. For Kimi-K2.5 it gives $L = 0.0030$, from one task (HumanEval/64) on which re-executed hidden tests pass but the released score is 0.

K.3 A FLOOR THAT SURVIVES EVERY STATE SHIFT

Proposition 3 certifies the aggregate floor under the law that generated $\mathcal{B}$, and Proposition 21(i) shows that only a cell-wise floor survives every reweighting of the states. Certifying the cell-wise floor directly combines the two. Call an action *sound* if $V_u \geq Y_B$ for every realization. Under the matched-outcome coupling of Appendix K.2, the reference action B is sound because $V_B = Y_B$, and the $\theta = 1$ replay actions, whose fallback is B, are sound by Proposition 21(i). Admissibility of g_B alone does not suffice, because a randomized g_B may play actions that are not individually sound.

Corollary 22 (Certified floor under every state shift)**.** *In the setting of Proposition 3(b), fix $\epsilon \geq 0$, and assume that the reference is a single action $B \in \mathcal{U}(x)$ in every state with $V_B = Y_B$ pointwise. Fix on $\mathcal{A}$, for each state x, an order of finitely many non-sound candidate actions. On the tasks of $\mathcal{B}$ in state x, test the candidates in that order with p-values $p_{x,u}$ that are valid at level $\alpha/|\mathcal{X}|$ for the null $\Delta(x,u) < -\epsilon$, stopping at the first non-rejection. Let the certified menu $\hat{\mathcal{U}}(x)$ contain the sound actions, which include B, and the candidates rejected before the stop. Then, with probability at least $1-\alpha$, every ex-ante router supported on $\hat{\mathcal{U}}$, including one chosen after seeing $\mathcal{B}$, satisfies $q(x) \geq -\epsilon$ in every state and hence $Q' \geq -\epsilon$ for every $P' \in \mathcal{P}_{\mathrm{state}}(P)$.*

Proof. Condition on $\mathcal{A}$ and on the states $x_1, \ldots, x_n$ of $\mathcal{B}$. The tasks of $\mathcal{B}$ in state x are then i.i.d. from $P(\cdot \mid x)$ and independent across states, so the Hoeffding and empirical-Bernstein p-values computed from them are valid; a state with fewer than two tasks certifies nothing. Within state x, the argument of Proposition 3(b) at level $\alpha/|\mathcal{X}|$ shows that the probability of certifying some candidate with $\Delta(x,u) < -\epsilon$ is at most $\alpha/|\mathcal{X}|$. A union bound over states bounds the probability of any such error by α. Outside this event, every certified action has $\Delta(x,u) \geq -\epsilon$ and every sound action has $\Delta(x,u) \geq 0 \geq -\epsilon$, so every router supported on $\hat{\mathcal{U}}$ has $q(x) \geq -\epsilon$ in every state. The menu $\hat{\mathcal{U}}(x)$ is nonempty because it contains B. Proposition 21(i) then gives the bound on Q'. Taking expectations removes the conditioning. □

What the robustness costs. On the X9 menu, the sound actions are Opus and the seven $\theta = 1$ replays. In each cell, the candidates are the non-sound actions with fitted $\Delta V \geq -\epsilon$ on $\mathcal{A}$, in decreasing order of that value, and the p-values are empirical Bernstein at level $0.05/21$. The router is the fitted ratio optimum on $\mathcal{A}$ over the certified menus. Over the same 200 unstratified splits, it keeps every test cell at $q(x) \geq 0$ in all 200 splits, at a median $2.34\times$ (2.33–2.35) Opus efficiency for every $\epsilon \in \{0, 0.005, 0.01\}$. Most of this gain comes from the $\theta = 1$ replays, whose acceptance is oracle-grade: the router that uses only sound actions already gives $2.00\times$. With direct dispatch alone, which needs no verifier, the certified menus give $1.99\times$ (1.99–1.99). On the seed split, the two routers give $2.34\times$ $[2.20, 2.51]$ with $\Delta V = +0.0915$ $[+0.0810, +0.1017]$ and $1.99\times$ $[1.88, 2.12]$ with $+0.0673$ $[+0.0576, +0.0767]$; the direct router dispatches away from Opus in only 2 of the 21 cells. By contrast, the aggregate certified router of Proposition 3, at a median $19.38\times$, is negative on the test set in a median of 8 of 21 cells, with median worst cell -0.217, and satisfies the cell-wise floor in none of the 200 splits.

K.4 Relation to prior work

Testing. The argument in Proposition 3(b) is the classical fixed-sequence procedure for a priori ordered hypotheses (Maurer et al., 1995; Westfall & Krishen, 2001). Each hypothesis is tested at the full level α until the first non-rejection, which controls the familywise error rate and hence the probability of selecting a violating router. Learn then Test (Angelopoulos et al., 2025) casts risk control as multiple testing over a parameter family and uses fixed-sequence testing when the risk is expected to be monotone in a scalar parameter. Bates et al. (2021) give the earlier construction with upper confidence bounds along a nested family. Selective classification with guaranteed risk (Geifman & El-Yaniv, 2017) and the cascaded selective evaluation of Jung et al. (2025), which escalates from weaker to stronger LLM judges, calibrate acceptance thresholds with finite-sample guarantees. The two-sample design with a fallback to a known policy is that of high-confidence policy improvement (Thomas et al., 2015) and of Seldonian algorithms (Thomas et al., 2019): a candidate is chosen on one part of the data, a safety test on the other part must show with high confidence that it is not worse than a baseline, and otherwise no new policy is returned. Corollary 22 splits α across states in Bonferroni fashion and tests in a fixed sequence within each state. We claim no new testing principle.

Optimization. Part (a) consists of parametric-programming facts about the fitted linear program. The price path is the solution path of its parametric objective $\hat{v} - \lambda \hat{c}$ (Gass & Saaty, 1955), and the two-router mixture of part (a)(3) is the one-constraint structure of constrained Markov decision processes (Beutler & Ross, 1985; Altman, 1999). Proposition 21(i) is the group-shift observation behind group distributionally robust optimization (Sagawa et al., 2020): the worst case over mixtures of groups is the worst group. Here it is applied to the quality change of a fixed router rather than to a training loss. Parts (ii) and (iii) specialize the cascade accounting of Eq. (5).

What is new here. The contribution is the choice of family, of certified quantity, and of fallback, which makes these tools fit the paper's accounting. The family is the price path of the fractional program of Theorem 2, not a grid over one threshold. It orders a vector of state-wise choices by one price, has at most $1 + \sum_x (|\mathcal{U}(x)| - 1)$ members, is strictly monotone in fitted value, cost, and efficiency, and its neighbour mixtures contain a fitted floor optimum at every tolerance. The certified quantity is the reference-relative quality change Q of Lemma 1, and the fallback is the reference itself. Certification and the in-sample price λ_ϵ therefore refer to the same object, and the output is a certified price $\hat{\lambda}$ and a data-chosen margin, which replace the price guess and the margin search. Unlike high-confidence policy improvement, which tests one candidate, the procedure tests the whole ordered path at level α and returns its most efficient certified member. Proposition 21 and Corollary 22 then separate three kinds of floor. An aggregate dispatch floor is a statistical property of the law that generated the certification sample. A cell-wise dispatch floor survives every reweighting of states, at a price we measure. A floor for verified escalation depends on the acceptance rule's false-accept loss on the target, which vanishes for every target when acceptance is reference-sound, for instance when it certifies the top score. None of the three carries an efficiency guarantee that holds on every target: efficiency transfers exactly when $Q' + \eta'_B \mathcal{S}' \geq 0$, which for escalation depends on how often the target accepts.

L Portfolio complementarity

L.1 Proof of Proposition 5

Write $\mathcal{V} = \mathbb{E}[V_S]$, $\mathcal{D} = \mathbb{E}[D_S]$, $\Delta\mathcal{V} = \mathbb{E}[V_{S^+} - V_S]$, and $\Delta\mathcal{D} = \mathbb{E}[D_{S^+} - D_S]$. Direct subtraction gives

$$\frac{\mathcal{V} + \Delta\mathcal{V}}{\mathcal{D} + \Delta\mathcal{D}} - \frac{\mathcal{V}}{\mathcal{D}} = \frac{\Delta\mathcal{V} - \eta(S)\Delta\mathcal{D}}{\mathcal{D} + \Delta\mathcal{D}}. \tag{60}$$

The denominator is positive, which proves the criterion. All execution, selection, and verification effects enter the paired values and costs. A requirement relative to the current portfolio is separately $\Delta\mathcal{V} \geq -\epsilon$; a floor relative to another fixed reference uses that reference's expected value.

To derive (25), let the exact selecting verifier give hit probabilities p and $p + r$. Write the current conditional means as v_h, v_{miss}, b and their enlarged-portfolio means as $v_h + \delta v_h, v_{\text{miss}} + \delta v_{\text{miss}}, b + \delta b$. Total expectation gives

$$\Delta\mathcal{V} = r(v_h - v_{\text{miss}}) + (p + r)\delta v_h + (1 - p - r)\delta v_{\text{miss}}, \tag{61}$$

$$\Delta\mathcal{D} = d_m(S) - rb + (1 - p - r)\delta b. \tag{62}$$

Substitution into (22) proves the shift correction. The δv_h term includes changes in the selected accepted answer; the remaining terms account for the changed fallback population. Common conditional means set all three shifts to zero and recover (24).

L.2 Proof of Proposition 6

Fix a portfolio S and let

$$p = p(S), \qquad \mathcal{V} = v_{\text{miss}} + p(v_h - v_{\text{miss}}), \qquad D = a(S) + (1 - p)b > 0.$$

For a candidate $m \notin S$, write

$$r = r_m(S) = \mathbb{P}\left(E_m \cap \bigcap_{i \in S} E_i^c\right), \qquad d = d_m(S) = a(S \cup \{m\}) - a(S).$$

The enlarged portfolio has expected value $\mathcal{V} + r(v_h - v_{\text{miss}})$ and positive expected cost $D + d - rb$. Subtracting efficiencies gives

$$\eta(S \cup \{m\}) - \eta(S) = \frac{r\{v_h - v_{\text{miss}} + \eta(S)b\} - \eta(S)d}{D + d - rb}. \tag{63}$$

The numerator gives the criterion; d includes changes in aggregation and verification cost. The exact selecting verifier realizes the portfolio's virtual best solver, and $r_m(S)$ is its component marginal contribution (Xu et al., 2012; Fréchette et al., 2016).

This calculation requires no independence among success events. It does require the same conditional values v_h, v_{miss} and fallback cost b for the two portfolios. When the fallback population or selected accepted answers change, (23) remains exact and (25) supplies the terms omitted by the homogeneous formula.

Static coverage is monotone submodular under any fixed joint law of the events (Appendix J.4); conditional marginals after observing outcomes need not satisfy the analogous adaptive property.

L.3 Proof of Proposition 7

Let $\mathbb{P}(E_0) = 1/2$. Set $(E_1, E_2) = (E_0, E_0^c)$ in one world and (E_0^c, E_0) in the other. Both worlds have marginal successes $(1/2, 1/2, 1/2)$ and candidate costs c. A marginal-only selector therefore chooses m_1 with the same probability q in both worlds. Its costs are $2c + qb/2$ and $2c + (1-q)b/2$, so

$$\max\left\{2c + \frac{qb}{2},\, 2c + \frac{(1-q)b}{2}\right\} \geq 2c + \frac{b}{4}.$$

An informed selector chooses the complement and pays $2c$. The perfect fallback gives every policy unit value. The result covers randomized selection before the primary's output; it does not cover learning dependence through additional observations.

Remark 23 (Identifiable range of residual coverage). For $S = \{m_0\}$, the Fréchet bounds are $\max\{0, p_m - p_0\} \leq r_m(S) \leq \min\{p_m, 1 - p_0\}$. Both endpoints are attainable by varying the overlap of E_m and E_0 while fixing their marginals. Thus marginals identify an interval, not the residual coverage itself.

L.4 Residual steering: setting and statement

A role or context view can change where a candidate allocates its effort. To isolate that effect, let a task's accepted answer type $T^\star$ have law q on a finite set $\mathcal{T}$, and let the two proposals $X_1 \sim p_1$ and $X_2 \sim p_2$ be independent of each other and of $T^\star$ conditional on the observable state. Fix a region $T_0 \subseteq \mathcal{T}$ to de-emphasize, chosen before the proposals are realized, and write $T_1 = T_0^c$. For $\rho \in [0, 1]$, define

$$p_2^\rho(t) = \frac{p_2(t)}{Z(\rho)} \begin{cases} \rho, & t \in T_0, \\ 1, & t \in T_1, \end{cases} \qquad Z(\rho) = \rho W_0 + W_1, \tag{64}$$

where $W_k = \sum_{t \in T_k} p_2(t)$. Lower ρ moves effort out of T_0. If $a_1 = \sum_t q(t)p_1(t) < 1$, the target law after primary failure is

$$q^-(t) = \frac{q(t)(1 - p_1(t))}{1 - a_1}, \qquad h(\rho) = \sum_t q^-(t) p_2^\rho(t).$$

Proposition 24 (Residual steering)**.** *Assume that $\mathcal{T}$ is finite; that, conditional on the observable state, $T^\star \sim q$, $X_1 \sim p_1$ and $X_2 \sim p_2$ are mutually independent; that $a_1 < 1$; that T_0 is fixed before the proposals are realized; and that $W_0, W_1 > 0$. For a real function f on $\mathcal{T}$ let $\bar{f}_k = W_k^{-1} \sum_{t \in T_k} f(t)p_2(t)$ be its p_2-weighted mean over T_k and $M_f(\rho) = \sum_t f(t) p_2^\rho(t)$. Then*

$$M_f'(\rho) = \frac{W_0 W_1}{Z(\rho)^2}\big(\bar{f}_0 - \bar{f}_1\big), \qquad \textit{in particular} \qquad h'(\rho) = \frac{W_0 W_1}{Z(\rho)^2}\big(\bar{q}_0^- - \bar{q}_1^-\big).$$

Hence reducing ρ strictly increases the conditional coverage $h = M_{q^-}$ if and only if $\bar{q}_1^- > \bar{q}_0^-$, and strictly reduces candidate 2's standalone quality $a_2(\rho) = M_q(\rho)$ if and only if $\bar{q}_0 > \bar{q}_1$; both can hold at once. Under an exact selecting verifier the pair succeeds with probability $a_1 + (1 - a_1)h(\rho)$.

This finite answer-type model isolates a distinction that marginal scoring misses. Applying Proposition 6 or differentiating (18) converts the coverage change into an efficiency comparison only when the compared values and costs satisfy their stated conditions; in particular, changing the failure set may change the fallback's conditional quality (Appendix L.6).

L.5 Proof of Proposition 24

Bayes' rule gives $q^-(t) = \mathbb{P}(T^\star = t \mid X_1 \neq T^\star)$. For any real f,

$$M_f(\rho) = \frac{\rho W_0 \bar{f}_0 + W_1 \bar{f}_1}{Z(\rho)},$$
$$M_f'(\rho) = \frac{W_0 W_1}{Z(\rho)^2}(\bar{f}_0 - \bar{f}_1).$$

Conditional independence identifies $M_{q^-} = h$ and $M_q = a_2$, proving both monotonicity statements. Exact selection gives $p_{\text{pair}} = a_1 + (1 - a_1)h$. With costs and values fixed,

$$\frac{d}{dp}\frac{v_{\text{miss}} + p(v_h - v_{\text{miss}})}{a + (1-p)b} = \frac{v_h(a+b) - v_m a}{[a + (1-p)b]^2},$$

so the numerator determines the efficiency effect. This is a static reweighting chosen before X_1; sequential adaptation to the realized proposal lies outside (64). Proposition 6 itself does not require proposal independence.

L.6 A four-type construction

The following finite construction separates individual quality from complementarity without estimation error. Take

$$q = (0.30, 0.20, 0.40, 0.10), \qquad p_1 = (0.45, 0.45, 0.05, 0.05),$$
$$p_2 = (0.40, 0.40, 0.10, 0.10), \qquad T_0 = \{1, 2\}.$$

Both members retain standalone quality $1/4$, while

$$\overline{q^-}_0 = \frac{11}{60} < \frac{19}{60} = \overline{q^-}_1, \qquad h(\rho) = \frac{0.110\rho + 0.0475}{0.60\rho + 0.15}.$$

Thus steering away from T_0 increases pair coverage. With each member costing one, fallback costing ten, and fixed $v_h = 1$, $v_{\text{miss}} = 4/5$,

$$\eta(\rho) = \frac{\frac{4}{5} + \frac{1}{5}p_{\text{pair}}(\rho)}{12 - 10p_{\text{pair}}(\rho)}.$$

Table 23 gives the resulting trade-off. The candidate $p_3 = (0.60, 0.40, 0, 0)$ has higher standalone

Table 23: The four-type construction: steered proposal law, standalone quality, conditional coverage, pair success, and efficiency at three steering levels.

ρ	p_2^ρ	a_2	$h(\rho)$	p_{pair}	η
1	$(.4, .4, .1, .1)$	$1/4$	$21/100$	$163/400$	$1763/15850$
$1/4$	$(.25, .25, .25, .25)$	$1/4$	$1/4$	$7/16$	$71/610$
0	$(0, 0, .5, .5)$	$1/4$	$19/60$	$39/80$	$359/2850$

quality $13/50$, but lower pair success $393/1000$ than even the unsteered pair's $163/400$. This establishes a limitation of marginal ranking within this pool, not a general advantage of steering over adding models.

The fixed-fallback assumption matters. In the same construction, instead let fallback succeed exactly on types 3 and 4. Final value becomes

$$v(\rho) = 0.725 + \frac{0.11\rho}{Z(\rho)}.$$

Efficiency $v(\rho)/[12 - 10p_{\text{pair}}(\rho)]$ falls from $167/1585$ to $29/285$ as ρ decreases from one to zero, despite increasing coverage: the fallback's conditional value changes with the residual task distribution.

M INFORMATION-STATE DYNAMICS

M.1 PROOF OF PROPOSITION 11

Index the first decision by $n = 0$. Arrivals are independent of the pre-decision store; expiration precedes a replacing write. A record of either type is freshly written or survives expiration without replacement, giving

$$i_{j,n+1} = \nu_j \omega_j + \alpha_j i_{j,n}, \qquad w_{j,n+1} = \nu_j \psi_j + \alpha_j w_{j,n}. \tag{65}$$

Their fixed points and empty-start solutions are

$$i_j^* = \frac{\nu_j \omega_j}{1-\alpha_j}, \qquad w_j^* = \frac{\nu_j \psi_j}{1-\alpha_j},$$
$$i_{j,n} = i_j^*(1-\alpha_j^n), \qquad w_{j,n} = w_j^*(1-\alpha_j^n).$$

These are (36). Using the record gains of (34) and conditioning on the arriving cell,

$$\begin{aligned}\mathbb{E}[V_n - \eta_{\mathrm{U}} D_n] &= \sum_j \nu_j \left[v_j^{\mathrm{U}} - \eta_{\mathrm{U}} c_j^{\mathrm{U}} + i_{j,n} g_j^{\mathrm{I}} + w_{j,n} g_j^{\mathrm{W}} \right] \\ &= \sum_j \nu_j G_j (1-\alpha_j^n),\end{aligned} \tag{66}$$

where, as in (35),

$$G_j = i_j^* g_j^{\mathrm{I}} + w_j^* g_j^{\mathrm{W}} = \frac{\nu_j(\omega_j g_j^{\mathrm{I}} + \psi_j g_j^{\mathrm{W}})}{1-\alpha_j}.$$

The uninformed-policy terms cancel after averaging over cells.

Let $\bar{c}_\infty > 0$ be the stationary expected cost. Passing to the limit in (66) gives the stronger comparison

$$\eta_\infty - \eta_{\mathrm{U}} = \frac{\sum_j \nu_j G_j}{\bar{c}_\infty}. \tag{67}$$

Summing (66) as a geometric series gives (37), an identity for expected aggregates.

If $g_j^{\mathrm{I}} > 0 > g_j^{\mathrm{W}}$, the sign of G_j is the sign of

$$\omega_j g_j^{\mathrm{I}} + \psi_j g_j^{\mathrm{W}} = r_j \left[g_j^{\mathrm{I}} - \frac{\psi_j}{r_j}(g_j^{\mathrm{I}} - g_j^{\mathrm{W}}) \right].$$

It is positive if and only if $\psi_j / r_j < g_j^{\mathrm{I}} / (g_j^{\mathrm{I}} - g_j^{\mathrm{W}})$.

For finitely many cells, convergence is geometric with factor at most $\max_j \alpha_j < 1$. Independent arrivals do not make the resulting hit sequence independent or Markov.

M.2 RECORDS INSIDE A VERIFIED CASCADE: PROOF OF PROPOSITION 12

The record changes only p_j^s; a_j, b_j, v_h, v_b are fixed. Thus

$$v_j^s = p_j^s v_h + (1-p_j^s) v_b, \qquad c_j^s = a_j + (1-p_j^s) b_j.$$

Subtracting the uninformed state, with $\Delta_j^s = p_j^s - p_j^{\mathrm{U}}$, gives

$$g_j^s = \Delta_j^s (v_h - v_b) + \eta_{\mathrm{U}} \Delta_j^s b_j = \Delta_j^s W_{\eta_{\mathrm{U}},j}, \qquad W_{\eta_{\mathrm{U}},j} = \eta_{\mathrm{U}} b_j + v_h - v_b.$$

If $W_{\eta_{\mathrm{U}},j} > 0$ and $\Delta_j^{\mathrm{I}} > 0 > \Delta_j^{\mathrm{W}}$, the common factor cancels:

$$\frac{g_j^{\mathrm{I}}}{g_j^{\mathrm{I}} - g_j^{\mathrm{W}}} = \frac{\Delta_j^{\mathrm{I}}}{\Delta_j^{\mathrm{I}} - \Delta_j^{\mathrm{W}}}.$$

This proves the noise threshold. When $W_{\eta_{\mathrm{U}},j} < 0$, increasing acceptance instead lowers gain.

Serving a record without verification. Serving an erroneous record of value zero directly at lookup cost c_ℓ gives

$$g_j^{\mathrm{W}} = \eta_{\mathrm{U}}(c_j^{\mathrm{U}} - c_\ell) - v_j^{\mathrm{U}},$$

which is nonnegative exactly when $v_j^{\mathrm{U}}/c_j^{\mathrm{U}} \le \eta_{\mathrm{U}}(1 - c_\ell/c_j^{\mathrm{U}})$. The saving can outweigh the lost value, so $g_j^{\mathrm{W}} < 0$ must not be assumed for an unverified served record.

Relation to verification. The write probabilities of Section E are parameters. They acquire a verification meaning under three hypotheses: (i) a record is written for the arriving cell if and only if the primary is accepted, so that $r_j = \mathbb{P}(H = 1 \mid \text{cell } j)$; (ii) the record is erroneous if and only if the accepted output is worse than the reference, $Y_P < Y_B$; (iii) as in the standing assumption of Section E, acceptance and the accepted output's quality relative to the reference do not depend on the record state, as when the records are written by attempts on which the router ignores the store (a logging or exploration stream) while the remaining rounds use the store without writing. Then

$$\frac{\psi_j}{r_j} = \mathbb{P}(Y_P < Y_B \mid H = 1,\ \text{cell } j).$$

A conditional reliability bound $\mathbb{P}(Y_P \ge Y_B \mid \mathcal{G}) \ge \tau$ on accepted writes, where $\mathcal{G}$ includes all selection information and cell identity, gives $\psi_j/r_j \le 1 - \tau$ (Appendix I.1). Thus $\tau > -g_j^{\mathrm{W}}/(g_j^{\mathrm{I}} - g_j^{\mathrm{W}})$ suffices under the signed-gain assumptions. This probability bound must be established for the accepted-write population; the numerical score threshold θ alone does not imply it. If the record changes acceptance and every accepted attempt writes, hypothesis (iii) fails and Remark 25 applies instead.

M.3 Writes that depend on the record

Remark 25 (State-dependent writes). Suppose the write type depends on the record the decision used: in record state $s \in \{\mathrm{U}, \mathrm{I}, \mathrm{W}\}$ the arriving cell j receives an informative write with probability ω_j^s and an erroneous one with probability ψ_j^s, with $r_j^s = \omega_j^s + \psi_j^s$, while expiration keeps its probability δ_j. Conditioning on the record state at the start of the round, the record probabilities $x_{j,n} = (i_{j,n}, w_{j,n})^\top$ obey the affine recurrence $x_{j,n+1} = A_j x_{j,n} + b_j$ with

$$A_j = \begin{pmatrix} \nu_j(\omega_j^{\mathrm{I}} - \omega_j^{\mathrm{U}}) + (1-\delta_j)(1 - \nu_j r_j^{\mathrm{I}}) & \nu_j(\omega_j^{\mathrm{W}} - \omega_j^{\mathrm{U}}) \\ \nu_j(\psi_j^{\mathrm{I}} - \psi_j^{\mathrm{U}}) & \nu_j(\psi_j^{\mathrm{W}} - \psi_j^{\mathrm{U}}) + (1-\delta_j)(1 - \nu_j r_j^{\mathrm{W}}) \end{pmatrix}$$

and $b_j = \nu_j(\omega_j^{\mathrm{U}}, \psi_j^{\mathrm{U}})^\top$. This follows by conditioning on the old record and substituting $\mathbb{P}(\mathrm{U}) = 1 - i_{j,n} - w_{j,n}$. If $\delta_j > 0$ and $\nu_j r_j^s < 1$ for every s, every state can reach U, which has a self-loop. The chain therefore converges to a unique stationary law, $x_j^* = (I - A_j)^{-1} b_j$. From an empty store, $x_{j,n} = (I - A_j^n) x_j^*$.

The gain calculation is unchanged: $\eta_\infty \ge \eta_{\mathrm{U}}$ exactly when $\sum_j \nu_j(i_j^* g_j^{\mathrm{I}} + w_j^* g_j^{\mathrm{W}}) \ge 0$. Under $g_j^{\mathrm{I}} > 0 > g_j^{\mathrm{W}}$, the cell threshold applies to the stationary erroneous share $w_j^*/(i_j^* + w_j^*)$, not a state-independent write fraction.

For ten equally likely cells, $r_j^s = 1/2$, $\delta_j = 1/100$, unit costs and values $(0.6, 0.9, 0.1)$, the gain threshold is $3/8$. State-independent erroneous fractions $e^{\mathrm{U}} = e^{\mathrm{I}} = e^{\mathrm{W}} = 0.3$ give gain $+0.050$. Keeping $e^{\mathrm{U}} = e^{\mathrm{I}} = 0.3$ but raising e^{W} to 0.6 leaves coverage unchanged while raising the stationary erroneous share to $0.401 > 3/8$ and changing gain to -0.018. Thus the base write error of 0.3 is insufficient to determine the sign once erroneous records reinforce erroneous writes.

Remark 26 (Accumulating stores). Keeping every record without expiration preserves coverage $1 - (1 - \nu_j r_j)^n$, but changes readout. With i.i.d. write types, the stored erroneous fraction converges to ψ_j/r_j. An idealized majority readout consequently becomes informative below $1/2$ and erroneous above $1/2$, whereas newest-record readout has threshold (38). For $g^{\mathrm{I}} = 0.3$, $g^{\mathrm{W}} = -0.5$, these disagree for error fractions in $(3/8, 1/2)$. Equations (36) and (37) concern newest-record readout.

M.4 Records that become stale

Section E lets records expire but never lets an informative record become erroneous, although a change in the capability pool or in the task distribution can invalidate a stored route. Suppose that in each

round, after the decision and the expiration step, an informative record that has survived expiration turns erroneous with probability $\vartheta_j \in [0,1)$, independently of everything else (a memoryless hazard), before the arrival's write; a record written in the round is fresh. With $\alpha_j = (1-\delta_j)(1-\nu_j r_j)$ as in (32) and $\beta_j = \alpha_j(1-\vartheta_j)$, the record probabilities obey

$$i_{j,n+1} = \nu_j\omega_j + \beta_j i_{j,n}, \qquad w_{j,n+1} = \nu_j\psi_j + \alpha_j w_{j,n} + \alpha_j\vartheta_j i_{j,n}.$$

The fixed points are

$$i_j^* = \frac{\nu_j\omega_j}{1-\beta_j}, \qquad w_j^* = \frac{\nu_j\psi_j + \alpha_j\vartheta_j i_j^*}{1-\alpha_j}, \qquad i_j^* + w_j^* = \frac{\nu_j r_j}{1-\alpha_j} :$$

staleness changes record type, not coverage. Under $g_j^{\mathrm{I}} > 0 > g_j^{\mathrm{W}}$, a cell contributes positively exactly when its stationary erroneous share

$$e_j^{\mathrm{eff}} = \frac{w_j^*}{i_j^* + w_j^*} = \frac{\alpha_j\vartheta_j + (\psi_j/r_j)(1-\alpha_j)}{1-\alpha_j+\alpha_j\vartheta_j}$$

is below $g_j^{\mathrm{I}}/(g_j^{\mathrm{I}} - g_j^{\mathrm{W}})$.

At $\nu_j = 0.1$, $r_j = 0.5$, $\delta_j = 0.01$, $g^{\mathrm{I}} = 0.3$, $g^{\mathrm{W}} = -0.5$ and write error $\psi_j/r_j = 0.2$, no staleness gives gain $+0.118$. A hazard $\vartheta_j = 0.02$ raises the stationary erroneous share to $0.392 > 3/8$ and gives gain -0.012; the sign changes at $\vartheta^* \approx 0.0177$. This stationary memoryless hazard does not describe a deterministic pool change at a fixed time.

M.5 An arbitrary reference and memory-maintenance overhead

The uninformed state of a router with memory can already incur a maintenance or logging cost. To compare with a reference that does not pay that cost, fix any $\lambda \in \mathbb{R}$ and define

$$\begin{aligned} J_U(\lambda) &= \sum_j \nu_j(v_j^U - \lambda c_j^U), \\ g_j^s(\lambda) &= (v_j^s - v_j^U) - \lambda(c_j^s - c_j^U), \qquad s \in \{I, W\}, \\ G_j(\lambda) &= \frac{\nu_j[\omega_j g_j^I(\lambda) + \psi_j g_j^W(\lambda)]}{1-\alpha_j}. \end{aligned}$$

The empty-start finite-horizon identity is

$$\sum_{n=0}^{N-1} \mathbb{E}[V_n - \lambda D_n] = N J_U(\lambda) + \sum_j \nu_j G_j(\lambda) \left[N - \frac{1-\alpha_j^N}{1-\alpha_j} \right]. \tag{68}$$

At stationarity the expected gain is $J_U(\lambda) + \sum_j \nu_j G_j(\lambda)$; with positive stationary expected cost, its sign is the sign of $\eta_\infty - \lambda$.

In particular, let a no-memory reference have the same mean value as the uninformed state, mean cost $c_0 > 0$, and efficiency η_0. If the uninformed state of the router with memory costs $c_0 + h$ with $h > 0$, then $J_U(\eta_0) = -\eta_0 h$. Memory is therefore at least as efficient as that reference if and only if

$$\sum_j \nu_j G_j(\eta_0) \geq \eta_0 h. \tag{69}$$

Here h is the difference in expected cost per round; any further record-dependent costs remain inside $g_j^s(\eta_0)$.

The proof is (66) with the uninformed term retained, followed by the same geometric sum.

M.6 Coverage under time-based and event-based retention

Consider first a record that remains usable for exactly E global rounds after it is written, with no other expiration. Assume E is a positive integer. Once the initial transient has passed, cell j is covered if

and only if at least one of the preceding E rounds wrote a record for it. The i.i.d. write probability per round is $\nu_j r_j$, and hence

$$\chi_j^{\mathrm{TTL}} = 1 - (1 - \nu_j r_j)^E, \qquad i_j^{\mathrm{TTL}} = \frac{\omega_j}{r_j}\chi_j^{\mathrm{TTL}}, \qquad w_j^{\mathrm{TTL}} = \frac{\psi_j}{r_j}\chi_j^{\mathrm{TTL}}. \tag{70}$$

Conditional on a write to j, its type is informative with probability ω_j/r_j. This remains true for the most recent such write, which proves the last two identities.

Now consider an event-window FIFO that retains the most recent M write events globally, including repeated entries for the same cell, and answers a query using the newest retained matching record. Put

$$\beta = \sum_j \nu_j r_j > 0, \qquad q_j = \frac{\nu_j r_j}{\beta}.$$

Discarding rounds with no write produces an i.i.d. sequence of cell marks with law q. Thus, after the FIFO has filled,

$$\chi_j^{\mathrm{FIFO}} = 1 - (1 - q_j)^M, \qquad i_j^{\mathrm{FIFO}} = \frac{\omega_j}{r_j}\chi_j^{\mathrm{FIFO}}, \qquad w_j^{\mathrm{FIFO}} = \frac{\psi_j}{r_j}\chi_j^{\mathrm{FIFO}}. \tag{71}$$

These are independent-reference caching formulas (Fagin, 1977). Replacing E by the mean span M/β underestimates FIFO coverage: Bernoulli's inequality gives $(1 - \beta q_j)^{M/\beta} \geq (1 - q_j)^M$. TTL fixes elapsed rounds while FIFO fixes write events. Capacity in distinct cells is different again (Corollary 27).

M.7 Proof of Proposition 13

Let $\kappa_z = 1/\zeta(z)$ and $\nu_j = \kappa_z j^{-z}$ for $j \geq 1$, where $z > 1$. Assume clean writes with a common probability $\omega > 0$ and a common positive informed-minus-uninformed hit gap Δ. By (70) with $r_j = \omega$, a cell is uncovered under a TTL of E rounds with probability $(1 - \omega\nu_j)^E$, so the TTL coverage deficit is

$$L_E = \Delta \sum_{j \geq 1} \nu_j (1 - \omega\nu_j)^E,$$

and from an empty store without expiration the uncovered probability at round n is $(1 - \omega\nu_j)^n$ by (36) with $\delta_j = \psi_j = 0$, so the per-round deficit is L_n (with $L_0 = \Delta$) and, summing the geometric series in n, the cumulative deficit is

$$R_N := \sum_{n=0}^{N-1} L_n = \frac{\Delta}{\omega} \sum_{j \geq 1} \big[1 - (1 - \omega\nu_j)^N\big]. \tag{72}$$

Occupancy form. Round n writes cell j with probability $\omega\nu_j$. If K_N counts distinct written cells, then (72) gives $R_N = (\Delta/\omega)\mathbb{E}[K_N]$, while $\omega L_N/\Delta$ is the expected unseen-cell write mass. The singleton identity $\mathbb{E}[K_{N,1}]/(\omega N) = L_{N-1}/\Delta$ also connects the deficit to Good-Turing missing mass (Good, 1953).

Rate and constants. The write-frequency counting function is $\bar{\nu}(x) = \lfloor (\omega\kappa_z/x)^{1/z} \rfloor \sim (\omega\kappa_z)^{1/z} x^{-1/z}$. The occupancy and missing-mass asymptotics of Karlin (1967) and Gnedin et al. (2007, Prop. 17 and Lemma 1), at index $1/z$, give

$$L_E \sim \frac{\Delta\kappa_z^{1/z}}{z}\Gamma\Big(1 - \frac{1}{z}\Big)(\omega E)^{1/z - 1}, \qquad R_N \sim \Delta\kappa_z^{1/z}\Gamma\Big(1 - \frac{1}{z}\Big)\omega^{1/z-1}N^{1/z}, \tag{73}$$

and hence the claimed rates.

A self-contained derivation of the constants. For a direct derivation, squeeze $(1 - x)^E$ between e^{-Ex} and $e^{-(1+\varepsilon)Ex}$ for sufficiently small x; the remaining finite head contributes exponentially little. For $f_c(x) = \kappa_z x^{-z} e^{-c\kappa_z x^{-z}}$,

$$\int_1^\infty f_c(x)\,dx \sim \frac{\kappa_z^{1/z}}{z}\Gamma(1 - 1/z)c^{1/z - 1}.$$

Unimodality bounds the sum-integral error by $O(c^{-1}) = o(c^{1/z-1})$, proving the first equivalent at $c = \omega E$. Applying the same squeeze to $1-(1-\omega\nu_j)^N$ gives $\mathbb{E}[K_N] \sim \Gamma(1-1/z)(\omega\kappa_z N)^{1/z}$; the monotone sum-integral error is at most one. This proves the second equivalent.

Relation to Proposition 11. With $\delta_j = \psi_j = 0$, constant ω, hit gap Δ and margin W_{η_U}, one has $1-\alpha_j = \omega\nu_j$ and $G_j = \Delta W_{\eta_U}$. Substitution in (37) gives shortfalls $W_{\eta_U} L_n$ per round and $W_{\eta_U} R_N$ cumulatively. Any persistent informed-versus-oracle gap contributes a separate linear term. Zipf arrivals alone do not determine the rate if Δ_j or ω_j varies with the cell.

M.8 Capacity measured in records

Corollary 27 (Capacity measured in records)**.** *Under the assumptions of Proposition 13, the expected number of cells holding a record under a TTL of E rounds is $m(E) = \sum_j [1-(1-\omega\nu_j)^E] \sim \kappa_z^{1/z}\Gamma(1-1/z)(\omega E)^{1/z}$, and the stationary deficit expressed through occupied capacity is*

$$L_E \sim \frac{\Delta\,\Gamma(1-1/z)^z}{z\,\zeta(z)}\, m(E)^{-(z-1)}.$$

A store holding at most M distinct cells, updated by the write events with least-recently-used replacement, obeys the same law with M in place of $m(E)$ for every $\omega \in (0,1]$ (Jelenković, 1999; Che et al., 2002), whereas the event FIFO of (41) *with M slots has deficit $\Theta(M^{-(z-1)/z})$.*

Proof. By (70) with $r_j = \omega$, cell j holds a record under the TTL if and only if it was written during the last E rounds, so the expected number of cells holding a record is

$$m(E) = \sum_j \big[1-(1-\omega\nu_j)^E\big] = \mathbb{E}[K_E] = \frac{\omega}{\Delta} R_E$$

by (72). Eliminating ωE between this asymptotic and (73) gives the TTL law.

Clean constant-probability writes have i.i.d. cell marks with law ν, regardless of ω. The LRU store therefore follows the independent reference model. Its cited fault asymptotic is $(1-1/z)\Gamma(1-1/z)^z$ times $\sum_{j>M}\nu_j \sim \kappa_z M^{1-z}/(z-1)$ (Jelenković, 1999), giving the same constant as TTL.

For event FIFO, (71) instead gives deficit $\Delta\sum_j \nu_j(1-\nu_j)^M$, equal to L_M at $\omega = 1$, hence exponent $(z-1)/z$. The M slots hold only $\mathbb{E}[K_M] = \Theta(M^{1/z})$ distinct cells; the remaining slots are duplicates. □

Remark 28 (Break-even horizon). With $C_z = \Delta\kappa_z^{1/z}\Gamma(1-1/z)\omega^{1/z-1}/z$, an empty-start deficit $L_N \sim C_z N^{-(z-1)/z}$ falls below a tolerance τ after $N^\star \sim (C_z/\tau)^{z/(z-1)}$ rounds. The exponent makes this horizon large when z is close to one; the rate is asymptotic, not a finite-capacity allocation guarantee.

N Extended Related Work

Model selection, cascades, and portfolios. Cascades and routers formalize parts of the problem (Chen et al., 2024b; Ong et al., 2025; Dekoninck et al., 2025); the acceptance step is a reject option (Chow, 1970), studied as confidence-based or budgeted deferral (Jitkrittum et al., 2023; Gupta et al., 2024; Aggarwal et al., 2024; Chen et al., 2020; Trapeznikov & Saligrama, 2013; Verma et al., 2023) and linked to speculative decoding (Narasimhan et al., 2025). Algorithm portfolios select complementary solvers per instance (Rice, 1976; Huberman et al., 1997; Gomes & Selman, 2001; Kotthoff, 2016): the exact selecting verifier of Section 3.2 is their virtual best solver, residual coverage is the marginal contribution of a component (Xu et al., 2012; Fréchette et al., 2016), and steering relates to the Hydra objective (Xu et al., 2010). Ensemble diversity and multi-proposer aggregation motivate combining outputs (Kuncheva & Whitaker, 2003; Brown et al., 2024; Jiang et al., 2023; Wang et al., 2025); boosting adapts component weights (Schapire, 1990; Freund & Schapire, 1997). Learned routers predict per-model quality from the query, through preference data, contrastive or graph representations, or model embeddings (Ding et al., 2024; Ong et al., 2025; Chen et al., 2024c; Feng et al., 2025; Zhuang et al., 2025), and BEST-Route also chooses how many samples to draw

(Ding et al., 2025); the quality-cost and kNN routers of Appendix A.4 are simple members of this family, and our accounting chooses where on such a frontier to operate. Our portfolio comparison asks when additional coverage justifies the complete calling cost.

Information, verification, and memory. Paying for a routing signal is a value-of-information and metareasoning question (Raiffa & Schlaifer, 1961; Howard, 1966; Russell & Wefald, 1991; Hay et al., 2014). Router memory relates to caches under the independent reference model (Fagin, 1977; Che et al., 2002), LLM response caches and model multiplexing (Bang, 2023; Zhu et al., 2023; Gaspar Schroeder et al., 2026), and erroneous agent memory (Chen et al., 2024e). The returns to additional calls and test-time compute depend on the task and allocation (Chen et al., 2024a; Snell et al., 2025), imperfect verifiers constrain resampling gains (Stroebl et al., 2026), and hedged execution trades expenditure for tail latency (Dean & Barroso, 2013). Feedback-driven self-repair (Chen et al., 2024d; Shinn et al., 2023) must be compared with resampling at equal cost (Olausson et al., 2024), as in our retry study, which also scores repairs on tests withheld from the feedback. These connections motivate measuring the decision improvement supplied by information together with its acquisition and retention costs.

Routing beyond model identity. Coordinators route worker instructions, roles, and contributions (Nielsen et al., 2026; Xu et al., 2026; Tang et al., 2026); harness and token-level routers condition on execution history or the generated prefix (Liu et al., 2026; Fu et al., 2026a); visual routers select models or token-pruning strategies (Yu et al., 2026; Liang et al., 2026); and state-transfer methods reuse internal representations or latent messages across models (Fu et al., 2026b; Heo et al., 2026; Qu et al., 2026; Zou et al., 2025). Lemma 1 applies to each as a complete-policy comparison of final value and total cost, but this paper evaluates only model routing, and ACRouter's persistent records (Zhou et al., 2026) are the only cross-task memory it studies.