CERTIFICATE STOPPING: EXACT VERIFIERS AND A PER-REQUEST COST CAP FOR LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

A service that bills per request must guarantee two things on every request: a correct answer and a bounded cost. A verifier, such as a proof checker, an instruction checker, a schema validator or a code test, can check an answer before delivery. Routers and cascades reduce cost by choosing which model answers, but they estimate quality, so a wrong estimate can deliver an incorrect answer, and they bound cost only on average, so no single request's cost is bounded.

We propose certificate stopping to close both gaps. A small model, the certifier, answers first and an exact verifier checks its answer; if accepted, the answer is delivered, otherwise the large model runs within the remaining budget, the cap minus what the certifier spent, which bounds every request's cost. The remaining question is what such a policy costs. The rule is simple, but its cost is not obvious: rejected requests are the hard ones, a rejection leaves the large model a smaller budget, and several certifiers draw on one budget. Our first theorem resolves the three effects exactly, as a cost identity whose terms are measured on an earlier run and which gives the cost ratio of a new large model before it runs when its cost is proportional to the old one's. Our second theorem gives the acceptance rate above which adding a certifier lowers the expected cost under the shared budget, with no assumption on costs or on dependence between certifiers; choosing the set exactly is NP-hard, so the order is fitted, not derived. We also bound the accuracy lost when the verifier may accept wrong answers. The policy requires no training.

On eight benchmark subsets whose pass criteria were fixed before their test requests were opened, the policy uses 0.274 to 0.654 of Qwen3-32B's parameter-weighted tokens and is 0.75 to 8.32 points more accurate, with a positive lower bound on seven, and three of them repeat under three seeds; the verifier there is the official scoring metric, so accepted answers pass by construction. Cost ratios predicted before large models from four further families ran meet the registered band on 23 of 24 cells, and a rule fixed in advance limits the claim to the small model's own family. No evaluated learned router or cascade is more accurate on held-out requests, and an uncapped control exceeds the cap on 212 of $10{,}040$ request evaluations, the policy on none. Where no verifier exists, the policy has no certificate to stop on, and on one official routing pool a trained cascade router is more accurate and cheaper than the policy.

1 INTRODUCTION

A service that answers requests with a language model and bills per request needs two guarantees on every request: a correct answer and a bounded cost. Neither is automatic. Answers can be wrong, and every request costs computation, more for larger models (Kaplan et al., 2020; Hoffmann et al., 2022). A service that always calls its largest model pays that cost on every request, including the many that a smaller model would have answered correctly; a service that calls a smaller model risks a wrong answer it cannot detect. The tension is per request: a guarantee that holds on average, good on most requests and bad on a few, is not what a caller billed per request receives.

Routing and cascading. Two families of methods reduce cost by choosing which model answers. A *router* chooses the model before seeing any answer (Ong et al., 2025; Ding et al., 2024; Zhang

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.

et al., 2025b); a *cascade* obtains a small model's answer and then decides whether to call a larger model, paying for both when it does (Chen et al., 2024a; Aggarwal et al., 2024; Dekoninck et al., 2025). Both leave both guarantees open: they decide from an estimate of answer quality, so a wrong estimate delivers an incorrect answer, and they constrain cost on average (Chen et al., 2024a; Dekoninck et al., 2025), so no single request's cost is bounded.

Checking the answer. The accuracy gap closes when a program can check the answer. A *public verifier* checks a stated property of an answer from the request and the answer alone, as a proof checker (de Moura & Ullrich, 2021), an instruction checker (Zhou et al., 2023), a schema validator or a code test does. An acceptance is a *certificate* of the checked property. The verifier is *exact* relative to the official metric if it never accepts an answer that the metric scores as incorrect, so an accepted answer is correct under that metric, while a rejection establishes nothing.

The remaining problem. Existing methods control expected cost, the probability of exceeding a budget (Valkanas et al., 2025), or the cost of an ensemble (Huang et al., 2025) (Appendix Table 5); none caps each request at the large model's own cost, and none stops on an exact verifier's acceptance. A cap raises a new question: all calls on a request share one budget, so a rejected cheap attempt leaves less for the large model, and the question is when the attempt is worth what it takes.

Our approach. *Certificate stopping* caps every request at the large model's own cost and stops on an exact verifier's acceptance. It tries small models, called *certifiers* because their accepted answers are certificates, returns the first answer that the verifier accepts, and otherwise runs the large model within the *remaining budget*, the cap minus the certifiers' spending. The policy requires no training; only the choice of certifiers uses requests set aside for fitting. A router or cascade may follow the last rejection; every measurement here uses the large model alone.

Contributions. The policy is simple; its cost is not obvious, because rejection selects the hard requests, shortens the large model and couples the certifiers through one budget. The theory resolves the three effects exactly: a cost identity with one term per effect (Theorem 1), its transfer to a new large model under proportional cost (Proposition 2), the sign of its selection term (Lemma 3), the acceptance rate above which one more certifier lowers the expected cost (Theorem 4), and the harm of an inexact verifier (Proposition 5).

1) *The cost ratio is predicted, within the certifier's family, before the large model runs.* The *cost ratio* divides the policy's average cost by the large model's average cost. Theorem 1 in Section 2 writes it as an identity in four terms measured on an earlier run of the certifier and the large model: the certifier's cost, the rejection rate, the large model's cost on rejected requests, and the shortening from the remaining budget. If a new large model's cost is proportional to the old one's on every request, the same terms give its cost ratio before it runs (Proposition 2). For large models of four further families, cost ratios predicted before those models ran fell inside a band fixed in advance on 23 of 24 *cells*, a cell being one benchmark subset paired with one certifier and one large model (Table 18 in Appendix F). The one miss overstates the policy's cost; a rule registered before those runs limits the claim to the certifier's own family (Section 4).
2) *Choosing several certifiers under one per-request budget.* With several certifiers, the policy must choose which to run and in what order; we propose no order rule; our result is about admission. Theorem 4 in Section 2 says adding a certifier lowers the expected cost if and only if its acceptance rate exceeds a threshold set by its cost share and its shortening of the large model, with no assumption on costs or on dependence between certifiers; with independent acceptances the whole set is a knapsack problem (Theorem 9, Appendix C), choosing it exactly is NP-hard (Theorem 16, Appendix H), and the measured policies choose their set on requests set aside for fitting (Section 3).
3) *Confirmed cost savings and their empirical boundaries.* On eight cells whose verifier is the official metric, the policy uses 0.274 to 0.654 of Qwen3-32B's parameter-weighted tokens. These cells are *confirmed* in the sense of Section 3, and the policy is 0.75 to 8.32 points more accurate than Qwen3-32B, with a positive lower bound on seven of the eight (Figure 2). No learned router or cascade evaluated is more accurate on those requests, none is at least as cheap and at least as accurate on any of 22 development cells, and an uncapped control with the same verifier is no more accurate and exceeds the cap on 212 of $10{,}040$ request evaluations, the policy on

none (Section 4). The claim stops where the verifier does: without a verifier the policy has no certificate to stop on, and on one official routing pool, cascade-routing (Dekoninck et al., 2025) is then more accurate than the policy and cheaper than the policy (Section 4).

Scope. The result of this paper is about cost, not capability. It holds where a public verifier can decide whether the certifier's answer may be delivered. It stops where no such verifier exists. Nothing here shows that composition beats the strongest available model on tasks no verifier covers. Even where a verifier exists, it decides one property of the answer. An acceptance establishes that property and nothing more.

2 CERTIFICATE STOPPING UNDER A PER-REQUEST BUDGET

An exact verifier removes the accuracy risk of stopping early, so what remains is cost. Three effects make the cost hard to predict (Figure 1). Rejected requests tend to be the hard ones, which cost the large model most. A rejection leaves the large model a smaller allowance, so its answer costs less but may be cut off before it is correct. Several certifiers draw on the same budget, so which to run and in which order cannot be decided separately. The same three effects appear in any cascade, retry loop or agent pipeline that caps each request's cost.

Definition 1 (Setting)**.** We consider the following objects (complete definitions: Appendix A).

1) *Requests and scoring.* A benchmark poses requests $x \in \mathcal{X}$, and its official metric scores an answer $y \in \mathcal{Y}$ by $u_x(y) \triangleq \mathbf{1}\{y \text{ is correct}\}$.
2) *Verifier.* A public verifier $v_x : \mathcal{Y} \to \{0,1\}$ reads only x and y and runs while the request is served. An acceptance $v_x(y) = 1$ is a *certificate*. The verifier is *exact* when it accepts only answers that the metric scores correct, $v_x(y) \le u_x(y)$ for all $y \in \mathcal{Y}$. Both maps are deterministic (Section 6).
3) *Actions.* An answer comes from an *action* a of the registered set $\mathcal{A}$: one call to one model with a fixed prompt, sampler and generation cap M_a, whose answer on x is $y_a(x) \in \mathcal{Y}$, written y_a when x is fixed, and whose *rejection set* is $R_a \triangleq \{x \in \mathcal{X} : v_x(y_a(x)) = 0\}$.
4) *What an action costs.* Let $I_a(x)$ be the prompt length of a on x and $O_a(x) \le M_a$ its output length. Its per-token charges w_a^{in} and w_a^{out} are the model's parameter count for an open checkpoint (parameter-weighted tokens, PWT) and the released prices for an API model (USD) (Chen et al., 2024a; Dekoninck et al., 2025). The *complete cost cap* $C_a(x) \triangleq w_a^{\text{in}} \cdot I_a(x) + w_a^{\text{out}} \cdot M_a$ is what a costs if it generates to its cap, so it is known before the call (Valkanas et al., 2025), and the *realized cost* $c_a(x) \triangleq w_a^{\text{in}} \cdot I_a(x) + w_a^{\text{out}} \cdot O_a(x) \le C_a(x)$ is known after it (Hu et al., 2024). For a cost cap $b \le C_a(x)$, $c_a(x; b)$ is the realized cost of a on x with its generation cap reduced so that its cost cap is b; thus $c_a(x; C_a(x)) = c_a(x)$. Both charges are per token, so metered cost is linear in the token counts (Kaplan et al., 2020; Hoffmann et al., 2022); Appendix G audits the FLOP cost. The verifier is not charged.
5) *Large model and certifiers.* The *large model* P is the action against which every cost and accuracy is measured, and the *certifiers* $a_1, \ldots, a_k$ are the other actions, small models whose answers the verifier checks, indexed in the order the policy runs them. Throughout, *large* and *small* refer to this cost.
6) *Budget.* A request may spend what one full call to P would cost, so its *per-request budget* is $B(x) \triangleq C_P(x)$. The *remaining budget* b starts at $B(x)$ and, after each call to a, is updated by $b \leftarrow b - c_a(x)$. After certifier a, P runs with $c'_P(x) \triangleq c_P(x; B(x) - c_a(x)) \le c_P(x)$.
7) *Policy and public state.* The policy reads the released context $\phi(x) \in \Phi$, the set of released contexts, before any call, and after each call to a it reads the answer y_a and the verdict $v_x(y_a)$; no label, judge score or model confidence enters. After calls to a set $\mathcal{A}' \subseteq \mathcal{A}$, the *public state* is the history $s \triangleq \big(\phi(x), (y_a, v_x(y_a))_{a \in \mathcal{A}'}\big) \in \mathcal{S}$. A *policy* is a map $\pi : \mathcal{S} \times \mathbb{R}_{\ge 0} \to \mathcal{A} \cup \{\text{stop}\}$ that, from the public state and the remaining budget, either names the next action or stops and delivers an answer; $\mathcal{A}_\pi(x) \subseteq \mathcal{A}$ is the set of actions that π calls on x, and $y_\pi(x) \in \mathcal{Y}$ is the answer that it delivers.
8) *What a policy must respect.* $C_a(x)$ is known before the call, so π calls a only if $C_a(x) \le b$. The total cost $c_\pi(x) \triangleq \sum_{a \in \mathcal{A}_\pi(x)} c_a(x)$ satisfies $c_\pi(x) \le B(x)$, because each call spends at most its cap, which is at most the remaining budget. A request with $c_\pi(x) > B(x)$ is a *violation*;

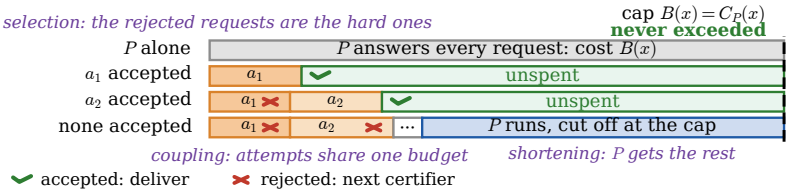


Figure 1: Three illustrative requests: cost as a fraction of the cap, one full call to the large model. A request stops at its first accepted answer (green: unspent); otherwise the large model runs in the rest of the cap.

every measurement replays recorded outputs, so such a request is not interrupted and is charged $c_P(x)$.

9) *Random request and averages.* Let X be a random request with distribution D on $\mathcal{X}$. Every function of X, such as $c_a(X)$, $c_P(X)$ and $v_X(y_a(X))$, is a random variable; $\mathbb{E}$ and $\Pr$ are the expectation and probability under D. When D is uniform on the requests $x_1, \ldots, x_n$ of one role, write $\mathbb{E}_n$ and $\Pr_n$: for $f : \mathcal{X} \to \mathbb{R}$ and $G \subseteq \mathcal{X}$, $\mathbb{E}_n[f(X)] \triangleq n^{-1} \sum_{i=1}^{n} f(x_i)$, $\Pr_n[G] \triangleq \mathbb{E}_n[\mathbf{1}_G(X)]$ and, when $\Pr_n[G] > 0$, $\mathbb{E}_n[f(X) \mid G] \triangleq \mathbb{E}_n[\mathbf{1}_G(X) \cdot f(X)] / \Pr_n[G]$. The *acceptance rate* of an action a is $q_a \triangleq 1 - \Pr[R_a]$ under the averaging in force, so $q_a = 1 - \Pr_n[R_a]$ on a role.
10) *Cost ratio and accuracy.* On the requests $x_1, \ldots, x_n$ of one role, the *cost ratio* of π is its total cost divided by the cost of answering the same requests with the large model alone, $\Gamma_n(\pi) \triangleq \mathbb{E}_n[c_\pi(X)] / \mathbb{E}_n[c_P(X)]$, and its *accuracy* is $\mathrm{Acc}_n(\pi) \triangleq \mathbb{E}_n[u_X(y_\pi(X))]$. □

The rule. Certificate stopping (Algorithm 1 in Appendix A) runs each certifier whose cap fits the remaining budget. It delivers the first answer that the verifier accepts, $y_\pi(x) = y_{a_{i^*}}$, where i^* is the least index such that a_i ran and $v_x(y_{a_i}) = 1$. If the verifier accepts no certifier, the policy replays the large model's recorded answer truncated at the remaining budget. Without an exact verifier nothing can be certified, and the policy follows a *root rule* that fixes one choice in advance for each released context $\phi(x)$. That choice is P, a cheaper registered action whose cap fits, or an *agreement cascade* of two cheap actions (Algorithm 2, Appendix A).

What the stopping decision guarantees. The optimal value $V(s, b) \in [0, 1]$ of recursion (1) in Appendix A.3 is the largest expected score any policy attains from public state s with remaining budget b, cost entering only through the budget. Certificate stopping stops exactly at the states that record a certified answer, with no threshold to tune, and that decision is optimal for (1) in expected score and in further cost (Proposition 6, Appendix A.3). This concerns the stop only; which certifiers to run, and in what order, (1) does not determine (Section 3 and the upcoming Theorem 4).[1]

Cost laws. The upcoming Theorem 1 is an identity between the averages $\mathbb{E}_n[\,\cdot\,]$ and $\mathbb{E}_n[\,\cdot \mid R_a]$ (Definition 1). It holds for any record set with no assumption, not even verifier exactness.

Theorem 1 (Cost identity)**.** *If certifier a runs on every request, so that P, when run, has remaining budget $B(x) - c_a(x)$ and cost $c'_P(x)$, and if $\Pr_n[R_a] > 0$ and $\mathbb{E}_n[c_P(X) \mid R_a] > 0$, then*

$$\Gamma_n = \alpha + (1 - q_a) \cdot \kappa \cdot \tau, \text{ where } \alpha \triangleq \frac{\mathbb{E}_n[c_a(X)]}{\mathbb{E}_n[c_P(X)]}, \ \kappa \triangleq \frac{\mathbb{E}_n[c_P(X) \mid R_a]}{\mathbb{E}_n[c_P(X)]}, \ \tau \triangleq \frac{\mathbb{E}_n[c'_P(X) \mid R_a]}{\mathbb{E}_n[c_P(X) \mid R_a]} \leq 1,$$

Here α is the certifier's cost relative to the large model's, the selection factor *κ is the cost of a rejected request relative to an average request, and the* shortening factor *τ is the fraction of that cost that the remaining budget leaves.*

Proof. See Appendix B.1. □

[1] We claim neither online realizability nor optimality over an unrestricted policy class. The analysis replays recorded outputs (Section 3) and does not establish that the policy can be served online as analysed, and Proposition 6 concerns the stopping decision at certified states only, not the choice among all policies that (1) admits.

Each factor is an average of logged costs and verdicts, so a deployed cascade reads from its logs which effect sets its cost (Table 3, Appendix C). The upcoming Proposition 2 turns the identity into a prediction for a large model that has not run.

Proposition 2 (Prediction under proportional cost)**.** *If a large model P' and a constant $\lambda > 0$ satisfy $c_{P'}(x) = \lambda \cdot c_P(x)$ for every request x and, separately, $c'_{P'}(x) = \lambda \cdot c'_P(x)$ for every $x \in R_a$, then certificate stopping with P' has cost ratio $\alpha/\lambda + (1 - q_a) \cdot \kappa \cdot \tau$, with α, q_a, κ and τ those of P.*

Proof. See Appendix B.2. □

This prices a change of large model from the logs of the current one before it runs; the hypothesis holds when P' generates as many tokens as P on every request at λ times the price per token, and Section 4 tests it on 24 cross-family cells with λ the parameter ratio.

The upcoming Lemma 3 signs κ when harder requests are both rejected more often and costlier.

Lemma 3 (Selection under monotone difficulty)**.** *For a latent difficulty Z of X, with $r(z) \triangleq \Pr[R_a \mid Z = z]$ and $m(z) \triangleq \mathbb{E}[c_P(X) \mid Z = z]$ the rejection probability and the large model's expected cost at difficulty z, respectively, $\mathrm{Cov}(\mathbf{1}_{R_a}(X), c_P(X) \mid Z) \geq 0$, $\Pr[R_a] > 0$ and $0 < \mathbb{E}[c_P(X)] < \infty$: if r and m are* comonotone*, $(r(z) - r(z')) \cdot (m(z) - m(z')) \geq 0$ for all z, z', then $\kappa = \mathbb{E}[c_P(X) \mid R_a]/\mathbb{E}[c_P(X)] \geq 1$, with equality only if $r(Z)$ or $m(Z)$ is a.s. constant.*

Proof. See Appendix B.3. □

In the common case that a certifier fails where the large model also generates longest, $\alpha + (1 - q_a)$ under-counts the cost and the measured κ must enter the budget; on the pooled development rows a conditional-cost predictor beats the rejection rate alone in both fold directions (Appendix E).

With several certifiers a policy chooses which to run and in what order; we propose no order rule (choosing the set exactly is NP-hard even with unit costs, and no polynomial-time algorithm beats the ratio $1 - 1/e$ unless $\mathrm{P} = \mathrm{NP}$; Theorem 16, Appendix H), and the policies of Section 4 fit their order on a finite library. Admission is what a shared budget adds: a certifier that runs leaves the large model less, and the upcoming Theorem 4 says when adding it lowers the cost.

Theorem 4 (Admission under a shared budget)**.** *On the requests reaching a certifier a with remaining budget $b(x)$, two policies are compared: without a, the large model runs at once under cost cap $b(x)$; with a, the certifier runs, the policy stops if a accepts, else the large model runs under cost cap $b(x) - c_a(x)$. With X the random request of Definition 1 conditioned on reaching a, so that $c_a(X)$ and $c_P(X; b(X))$ are the certifier's and the capped large model's costs, respectively, we define*

$$\rho \triangleq \frac{\mathbb{E}[c_a(X)]}{\mathbb{E}[c_P(X; b(X))]}, \qquad e \triangleq \frac{\mathbb{E}[c_P(X; b(X))] - \mathbb{E}[c_P(X; b(X) - c_a(X)) \mid R_a]}{\mathbb{E}[c_a(X)]},$$

the certifier's cost share *and* truncation elasticity*, respectively. If $\mathbb{E}[c_P(X; b(X))] > 0$ and $1 - \rho \cdot e > 0$, then adding a lowers the expected cost if and only if $q_a > \rho \cdot (1 - e)/(1 - \rho \cdot e)$.*

Proof. See Appendix B.4. □

On the eleven sealed cells of Table 4 in Appendix C, Theorem 4 is a fit, not a test. Its threshold uses three fit-role averages, so adding a certifier to a running cascade is decided before it is tried in production. With independent acceptances and constant costs the whole set is a knapsack problem (Theorem 9, Appendix C), and independence is optimistic: with rejections conditionally independent given a latent difficulty and nondecreasing in it, a set's realized failure probability is at least the independent model's (Lemma 8, Appendix C); Theorem 4 conditions on the requests reaching the certifier, so no such assumption enters it.

The upcoming Proposition 5 bounds the harm of an inexact verifier.

Proposition 5 (Harm of an inexact verifier)**.** *For a certifier a, with $V_a \triangleq v_X(y_a(X))$ and $U_a \triangleq u_X(y_a(X))$ the verdict and the correctness, respectively, of its answer on the random request, $q_a = \Pr[V_a = 1]$ its acceptance rate (Definition 1) and $p_a \triangleq \Pr[U_a = 1 \mid V_a = 1]$ its* precision*, a* harm*, a wrong certified answer delivered where the large model's answer is correct, has probability at most $q_a \cdot (1 - p_a)$, which is zero for an exact certificate ($p_a = 1$).*

Proof. See Appendix B.5. □

The bound needs no run of the large model: q_a is measured without labels, and p_a on the accepted requests of a labeled certifier pilot; Section 4 stops a pilot at the predicted harm rate and tests the bound on public-test code certificates. The bound covers false certificates only, that is, accepted answers that are wrong; a large model cut off after a rejection is a second harm, not bounded here, which the ledgers of Section 4 count.

3 EVIDENCE PROTOCOL

A *confirmed* number means one thing: a frozen policy was run once on requests it had never seen and passed a criterion written before they were opened. Every policy here is chosen from data among many candidates (Appendix A), so a number computed on the choosing requests is optimistic (Hastie et al., 2009; Cawley & Talbot, 2010; Dwork et al., 2015). The protocol therefore separates choosing from judging, as a confirmatory trial does (ICH, 1998): the requests that fix a policy's design are disjoint from those that score it, the scoring set is read once (Blum & Hardt, 2015), and the pass criterion is fixed before it opens (Nosek et al., 2018).

Roles. A *cell* is one benchmark subset paired with one certifier and one large model. A fixed public function of the request text splits its requests into four disjoint *roles*, used in this order. The *fit* role chooses the policy from a finite library fixed in advance and freezes it. The *mechanism* role tests the frozen policy against the gates below, on all of its requests and on each of two fixed halves, so that a pass cannot rest on one half; the *validation* role is read once, to decide whether to proceed. The *confirmation* role, opened at most once per registered policy and comparator and read only after a sealed authorization has fixed the configuration (Appendix A.4), yields every confirmed number; if its gate fails, the cell is a frozen negative, thereafter served byte for byte as the large model alone.

Two gates, declared before the confirmation role opens and never relaxed, decide a cell.

Definition 2 (Signed advantage and gates)**.** On a role of n requests, the *signed advantage* on request i is $\Delta_i \triangleq u_{x_i}(y_\pi) - u_{x_i}(y_P) \in \{-1, 0, +1\}$, a *harm* for $\Delta_i = -1$, a tie for $\Delta_i = 0$ or a *rescue* for $\Delta_i = +1$, with n_+ rescues and n_- harms. Its average over the n requests is the policy's accuracy minus the large model's on them, and ℓ is an exact lower bound on that average at confidence $1 - \delta$, obtained from Clopper–Pearson bounds on the rescue and harm rates (Definition 10, Appendix A.5). Thus $\ell \geq -1$ point certifies, at confidence $1 - \delta$, that the policy loses at most one accuracy point, and $\ell > 0$ that it gains. A cell passes the *efficiency gate* when (1) $\Gamma_n \leq 0.70$ on the role and on each half, $\ell \geq -1$ point at $\delta = 0.05$, and (2) no request is a violation. It passes the *improvement gate* when (1) the average of Δ_i is positive on each half, $\ell > 0$, (2) the policy's error is at most 0.90 times the strongest fixed action's, and (3) the rescues significantly outnumber the harms by the exact McNemar test (McNemar, 1947) with Holm correction (Holm, 1979) over the family of cells tested together, so that one lucky cell among many cannot pass. □

Status and counts. A cell is *confirmed* when its confirmation role passed the gate named in its sealed authorization. Every other number is *development*, a *frozen negative*, or a *diagnostic* run after labels were seen. Table 6 in Appendix E lists every cell: 23 confirmations, 14 distinct cells and 9 confirmations of the same frozen policy against a second large model on the same requests, not independent draws and uncorrected against their 32B counterparts; the Stanford Natural Language Inference cell, confirmed under an earlier gate of its own (Appendix I), is not counted.

4 EXPERIMENTS

Setup. Every benchmark is scored by its official checker: the Lean 4 kernel (de Moura & Ullrich, 2021) for MiniF2F (Zheng et al., 2022) and ProofNet (Azerbayev et al., 2023); instruction checkers for IFEval (Zhou et al., 2023), M-IFEval (Dussolle et al., 2025), Multi-IF (He et al., 2024) and IF-Bench (Pyatkin et al., 2025); a schema check for JSONSchemaBench (Geng et al., 2025); tests for LiveCodeBench (Jain et al., 2025) and MBPP (Austin et al., 2021); and the official metric for SimpleQA (Wei et al., 2024), HEADLINES (Sinha & Khandait, 2021) and SNLI (Bowman et al., 2015). LLMRouterBench (Li et al., 2026) has two settings, Performance-20 (open checkpoints, PWT) and

Performance-Cost (API models, released prices, USD). Models are from the Qwen3, Qwen2.5, Goedel-Prover-V2, phi-4, Mistral Small, Yi and Nemotron families (cited in Appendix I); the confirmed cells' certifiers range from Qwen3-0.6B (IFEval-es) to Goedel-Prover-V2-8B (MiniF2F).

With an exact verifier, the policy is confirmed cheaper and more accurate on every held-out role. One frozen policy, Qwen3-4B as certifier and Qwen3-32B as the large model, was scored on eight held-out confirmation roles: five Multi-IF cells, IFBench's classic constraints and both JSONSchemaBench collections, of 495 to $1{,}166$ requests. All eight are confirmed, with cost ratio Γ_n from 0.274 to 0.654 and 0.75 to 8.32 points more accuracy (Figure 2a). Seven cells pass both gates of Definition 2, each with at most six harms, all from the large model being cut off by the remaining budget rather than from a wrong certified answer; the eighth, JSONSchemaBench's other collections, has ℓ of -0.58 and passes the efficiency gate alone. Three of the roles, Multi-IF turn 1, IFBench classic and JSONSchemaBench GitHub, re-run under three seeds, repeat every preregistered gate with zero violations, at cost ratios 0.4296, 0.3047 and 0.3656 and exact lower bounds ℓ of $+5.90$, $+3.84$ and $+1.11$ points (Appendix E). Under a proof checker, the 128-task MiniF2F cell (Lean kernel) is confirmed at $+4.69$ points and cost ratio 0.443 with six rescues and no loss (Figure 2b); the larger held-out MiniF2F-test cell gains $+4.92$ points at cost ratio 0.692 with 2 truncation harms, but on one half its cost ratio is 0.704, above the ceiling, and its ℓ changes sign with the split of δ: a development row.

The same policy is confirmed against a 72B large model on nine of ten subsets. Unchanged, the policy then faces a larger large model: when Qwen2.5-72B replaces Qwen3-32B (Figure 2a), nine of the ten subsets are confirmed, with gains from $+1.50$ points at cost ratio 0.209 to $+10.29$ at 0.512. The tenth, Multi-IF turn 3 in five languages, fails confirmation at cost ratio 0.715 despite $+9.75$ points: the gain is there but the cost exceeds the ceiling.

A best-of-k sampling control matches the policy only under the proof checker. The natural alternative to calling the large model is to sample the certifier again, up to k times under the same budget, stopping at an answer the verifier accepts (Brown et al., 2024). On the development roles, the cascade certifies more requests than this control on the four instruction and schema cells but **loses on MiniF2F**: there the control matches the cascade's accuracy already at $k{=}4$, at cost ratio 0.071 against the cascade's 0.533 (Table 14, Appendix F): for proofs, sampling the prover beats calling the large model. On Multi-IF the control breaks the per-request budget on 38 requests, the cascade on none; on MiniF2F its violations were not measured.

Without a verifier, a fixed choice per context is confirmed on two contexts. Without a verifier the root rule of Section 2 fixes one action per context. On the Performance-Cost pool one cell passed validation and is confirmed on its 1,291 held-out SimpleQA requests. Qwen3-235B-A22B-2507 replaces GPT-5 (OpenAI, 2025) at 54.84% against 47.71% ($+7.13$ points, lower bound $+2.83$) at 0.004 of its released price (Figure 2b). Because the policy selects one action on every request, this result certifies model selection under the per-request budget, not composition; with no verifier, the guarantee is the cell-level bound only. On Performance-20 the root rule fixes a coder model as the action for the MBPP context and is confirmed there, at $+12.79$ points and cost ratio 0.055; on the pool's other eight contexts it keeps the large model.

Composition fails when acceptance is rare or the certificate is inexact. When frequent acceptance or an exact certificate is missing, composition fails. *Rare acceptance makes the policy too expensive.* On ProofNet the 7B prover is accepted on only 18.1% of statements, so the 32B prover runs on 82% of them, and the cost identity of Theorem 1 gives a cost ratio of 0.888, above the 0.70 ceiling; the cell is a frozen negative, a failed gate that is never retried (Section 3), despite $+11.54$ points and zero harms. Two more cells fail the same ceiling against the 32B large model: Multi-IF turn 3 in five languages ($+4.29$ points; cost ratio 0.676 overall but 0.715 on one half) and IFBench's OOD constraints ($+5.13$ points at 0.712). Figure 3a places the eleven 72B cells by acceptance rate and realized cost ratio, a description of the sealed record rather than a prospective test. *An inexact certificate can do harm.* On LiveCodeBench the public tests certify an answer while the hidden tests decide correctness, so certificates can be wrong. There the harm bound $q_a \cdot (1 - p_a)$ of Proposition 5 holds overall and on each difficulty stratum, and its slack widens with difficulty, both registered in

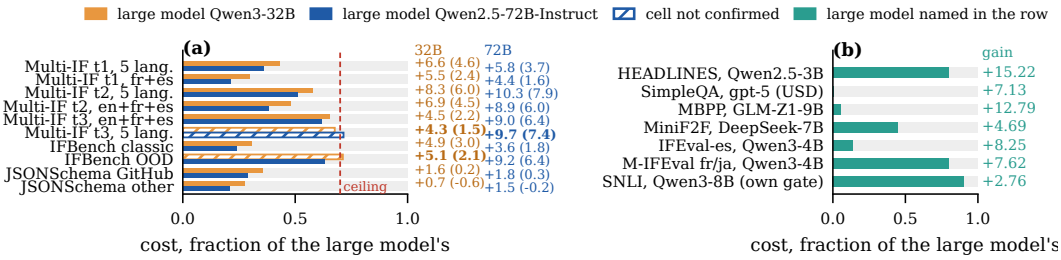


Figure 2: Confirmed cells. (a) One frozen policy against Qwen3-32B and Qwen2.5-72B on ten subsets; each label is the accuracy gain in points and, in parentheses, its exact 95% lower bound; hatched bars are not confirmed; (b) other confirmed cells (Appendix I).

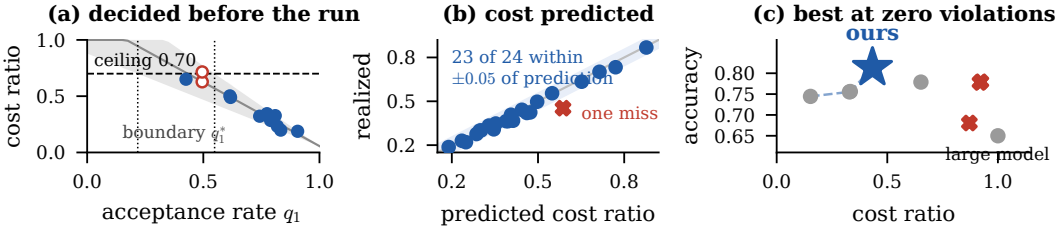


Figure 3: (a) The eleven 72B cells by acceptance rate q_1 and mechanism cost ratio; the band is the cost identity over the observed range of $\kappa \cdot \tau$, so the 0.70 ceiling is met only above a boundary in q_1, known before the large model runs; ticks join predicted to realized ratios and the open point is the frozen cell. (b) Predicted against realized cost ratio, 24 cross-family cells. (c) Averages over 22 development cells (Table 21, Appendix F); grey circles respect the cap, red crosses break it, uncapped control not plotted.

advance (Table 17, Appendix F), on a pilot's open fit role, stopped by the bound at a predicted harm rate of 0.087; the diagnostic reopens no frozen cell and awards no certificate.

Cost ratios were predicted before the large model ran, within family. Every cost ratio in both ladders was computed from the 32B run by Proposition 2, with λ the parameter ratio, and registered before the new large model ran. On the 24 cross-family cells, 23 realized ratios fall inside the ± 0.05 band named in advance (Figure 3b). The one miss is Multi-IF in five languages against phi-4, predicted 0.5875 and realized 0.4532. The rule registered for such a miss limits the claim to within-family predictions, since $\kappa \cdot \tau$ describes one large model's generation lengths; a certifier ablation shows that carrying $\kappa \cdot \tau$ across certifier sizes over-predicts by up to 0.1073 (Appendix I). Admitting the 32B as a middle stage follows Theorem 4's threshold, a fit derived after sealing on 11 two-cascade cells and out of sample on 24 of 24 cross-family cells (Appendix F). The cross-family ladder, phi-4, Mistral-Small-24B, Yi-1.5-34B and Nemotron-70B on six subsets (Table 18, Appendix F), is development evidence only: *its sealed authorizations allowed the mechanism role but no confirmation role, so a passing gate there adds no certified claim*. Of its 24 cells, 21 pass their gate with no cap violation; the three failures are all on IFBench's OOD subset, two predicted as ceiling failures.

No router or cascade matches the policy with a verifier; without one, cascade-routing beats it. Does the guarantee cost accuracy a router would find? Nine alternatives, a FrugalGPT-style cascade (Chen et al., 2024a), majority voting, six learned routers and AvengersPro (Zhang et al., 2025b), run under the same budget and ceiling on the development roles of both routing pools, scored by macro accuracy, the unweighted average of the per-context accuracies (Li et al., 2026), with per-request paired bounds, charged nothing for their own forward pass (or, for AvengersPro, its 7B embedding call), favouring them (Table 9, Appendix E). *With an exact verifier:* the ordering reverses (Figure 3c): on none of the 22 cells with a verifier is any alternative both at least as cheap and at least as accurate as the policy, and on the eight confirmation roles no learned router or cascade is more accurate than the policy on any of the 56 alternative-cell pairs; the seven cheaper zero-violation alternatives are

7.57 to 8.83 points less accurate (Tables 22 and 23, Appendix F). Two break the cap: AvengersPro on 701 certificate-cell requests, matching the policy on no cell (Appendix I), and majority voting on 114. The uncapped control is our policy with the large model run to its full cap after a rejection and no middle stage: accuracy 0.8140 at cost ratio 0.432 against the policy's 0.8135 at 0.433, but it breaks the per-request budget on 212 of the $10{,}040$ request evaluations over the 22 cells (2.11%), the policy on none; the cap costs 0.05 accuracy points. *Without a verifier:* on the open-checkpoint pool, where it never stops early, **cascade-routing beats it** on both readings: $+3.48$ macro points, and per request $+3.63$ with exact lower bound $+0.44$, at cost ratio 0.851 and zero violations. That router fails the cost ceiling ($0.851 > 0.70$) and owes no per-request guarantee, but our policy calls no verifier there and has nothing to stop on. On that pool the paired lower bounds of our fit-selected and mechanism-gated root policies are -1.74 and -1.01 points. Two of the four predictions registered for this comparison did not hold (Appendix F).

What the record does not establish. Preregistration obliges reporting every registered attempt, including failures (Nosek et al., 2018; Schulz et al., 2010), and the attempts below failed. Planning problems (the RCPSP pools) reached validation only and certify nothing. Refining an answer through a tree of further calls was swept on Performance-20; no tree was selected over the large model on any accuracy cell, and the coder-root MBPP cell confirms one context only. A hybrid that served verifiable requests by certificate stopping and the rest by a learned router at the unspent budget failed its gate (Appendix F). All frozen negatives stay.

5 Related work

Routers and cascades. Routers (Ong et al., 2025; Ding et al., 2024; Zhang et al., 2025b) and cascades (Chen et al., 2024a; Aggarwal et al., 2024) estimate answer quality and bound cost on average. Cascade-routing (Dekoninck et al., 2025) chooses the next model and the stop jointly with test checks but optimizes expected cost; ThriftLLM (Huang et al., 2025) bounds the cost of a chosen subset of an ensemble; C3PO (Valkanas et al., 2025) bounds the probability of exceeding a budget; CP-Router (Su et al., 2026) routes by conformal uncertainty; TREACLE (Zhang et al., 2024), RLCascadeRouter (Huang et al., 2026) and Router-R1 (Zhang et al., 2025a) learn policies. Certificate stopping differs on both guarantees: it stops only on an exact certificate, so a delivered small-model answer is correct, and it caps every request at the large model's own cost. Only cascade-routing is compared against here. Verifiers appear in prior work in a different role: verifier-guided generation (Li et al., 2022), self-debugging (Chen et al., 2024b) and self-consistency (Wang et al., 2023) use a checker to pick the best of several samples from one model, and selective prediction (Chow, 1970) abstains instead of answering. Here the verifier decides a different question, whether a cheaper model's answer may be delivered in place of the large model's, under a per-request budget.

6 Limitations and conclusion

Limitations. Of the 21 routing contexts only Performance-20's MBPP has a public verifier, and the results certify cost under it, not quality that the checker does not measure. Finite cells imply no population guarantee: each confirmed cell is one draw, the claim is within family, and a larger model is not automatically a better large model. Metered cost is not latency or memory. We report nothing under concurrency because we did not measure it. A control from the certifier's family, selected like the policy, is more accurate than the policy on two confirmation roles, by 1.01 points on Multi-IF at 1.06 times its tokens and by 0.78 points on IFBench at 0.97 times. The preregistered 0.70 ceiling, the checker defects, and the API cells, which certify those models on those dates and support no universal model-quality or scaling claim, are detailed in Appendix I.

Conclusion. An exact verifier turns a small model's accepted answer into a certificate, and a per-request cap turns the large model into a bounded fallback; certificate stopping combines the two. On eight confirmed cells it runs at $\Gamma_n \in [0.274, 0.654]$ and is more accurate than the large model, its cost ratio was predicted within family before the large model ran, and no router or cascade evaluated is at least as cheap and at least as accurate. Where no verifier exists the policy has nothing to stop on, and on one pool a router that owes no guarantee beats it, which is the boundary this paper locates.

Ethics statement

This work evaluates released language models on public benchmarks under their own licences and official checkers at fixed versions. It involves no human subjects, no new data collection, no personally identifiable information, and no deployment to users; we release no model weights. The method's mechanism is a certificate, and a certificate that is not exact can deliver a cheap answer with the appearance of verification, which suppresses the escalation that would have caught it. This is why the paper bounds the harm of an inexact certificate (Proposition 5), uses that bound to refuse a task family before spending compute on it, and recommends the policy only where the verifier is exact and public. The intended effect is to reduce metered inference cost, reported as parameter-weighted tokens or released prices; we make no energy or environmental claim. Failed and frozen cells consumed a real share of the compute and are reported with the successful ones.

Reproducibility statement

Every number in the paper is computed from a sealed record that the supplement ships. Before any model output exists for a cell, a written authorization fixes the role assignment, the model checkpoints, the per-request cap, the boundary stride, the base seed, the official checker files, the gates and the excluded rows, and generation, scoring and replay refuse to run against anything that differs from it. A cell that fails its gate is frozen with the large model alone replayed byte for byte, and the confirmation role of a benchmark subset is opened at most once per registered policy and large model. The replays and analyses run on the sealed record with no GPU: the reference points of Table 13, the admission analysis of Table 4, the cost-identity predictions and panel (a) of Figure 3 are exact arithmetic over recorded per-request outcomes and costs. Reproducing the scoring additionally requires the official checker trees at the authorized versions, which the supplement names; for JSONSchemaBench the scorer asserts a fingerprint of its own runtime and stops with an error on a mismatch. Reproducing the generation requires the released checkpoints and a GPU; we give the model revisions, the decoding pins (greedy, repetition penalty fixed at 1.0, an explicit end-of-sequence rule), the shard counts and the cluster job scripts. Sampled generation is reproducible only on the same GPU architecture, so each cell names the cluster that generated it, and the API-pool cells reproduce exactly from the benchmark's released records but cannot be regenerated without access to the corresponding APIs.

The supplement ships the code, the authorizations, the sealed records behind every table and figure, and a manifest that binds each record to the number it supports; the paper itself prints no record identifiers. Absolute paths inside the records identify the authors, so identifying strings are replaced by placeholders; the manifest records the original and the shipped digest of every file and declares each redacted file as such, and no file differs from its original for any other reason.

Statement on the use of AI assistants

AI assistants were used to aid and polish writing, retrieve and discover related work, assist research ideation and execution, draft manuscript sections and LaTeX, and assist in developing and checking mathematical claims and proofs. The experimental design, the authorizations, the sealed artifacts, every reported number and every proof were written or checked by the authors, and no output of an AI assistant was used as experimental evidence beyond the registered benchmark models that the experiments evaluate.

REFERENCES

Marah Abdin et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W. Ayers, Dragomir Radev, and Jeremy Avigad. ProofNet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*, 2023.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1006–1014, 2015.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

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.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024a.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *International Conference on Learning Representations*, 2024b.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16 (1):41–46, 1970.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. ISSN 0006-3444.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction—CADE 28*, volume 12699 of *Lecture Notes in Computer Science*, pp. 625–635. Springer, 2021.

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 (ICML)*, volume 267 of *PMLR*, 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 *International Conference on Learning Representations (ICLR)*, 2024.

Antoine Dussolle, Andrea Cardeña Díaz, Shota Sato, and Peter Devine. M-IFEval: Multilingual instruction-following evaluation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6176–6191, 2025.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. JSONSchemaBench: A rigorous benchmark of structured outputs for language models. *arXiv preprint arXiv:2501.10868*, 2025.

T. E. Harris. A lower bound for the critical probability in a certain percolation process. *Mathematical Proceedings of the Cambridge Philosophical Society*, 56(1):13–20, 1960.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. Multi-IF: Benchmarking LLMs on multi-turn and multilingual instructions following. *arXiv preprint arXiv:2410.15553*, 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén

Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2): 65–70, 1979.

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. In *ICML Workshop on Agentic Markets*, 2024.

Keke Huang, Yimin Shi, Dujian Ding, Yifei Li, Yang Fei, Laks Lakshmanan, and Xiaokui Xiao. ThriftLLM: On cost-effective selection of large language models for classification queries. *Proceedings of the VLDB Endowment*, 18(11):4410–4423, 2025.

Shihong Huang, Shengjie Wang, Hong Ma, and Zhou Xu. RLCascadeRouter: Quality-estimator-free cascade routing via reinforcement learning. *arXiv preprint arXiv:2608.15817*, 2026.

ICH. ICH harmonised tripartite guideline E9: Statistical principles for clinical trials. Technical report, ICH, 1998.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, Zhen Wang, and Shuyue Hu. LLMRouterBench: A massive benchmark and unified framework for LLM routing. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022.

Yong Lin, Shange Tang, Bohan Lyu, Ziran Yang, Jui-Hui Chung, Haoyu Zhao, Lai Jiang, Yihan Geng, Jiawei Ge, Jingruo Sun, Jiayun Wu, Jiri Gesi, Ximing Lu, David Acuna, Kaiyu Yang, Hongzhou Lin, Yejin Choi, Danqi Chen, Sanjeev Arora, and Chi Jin. Goedel-Prover-V2: Scaling formal theorem proving with scaffolded data synthesis and self-correction. In *International Conference on Learning Representations (ICLR)*, 2026.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

Mistral AI. Mistral Small 3. `https://mistral.ai/news/mistral-small-3`, 2025.

L. G. Mitten. An analytic solution to the least cost testing sequence problem. *Journal of Industrial Engineering*, 11:17, 1960.

Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. In *International Conference on Learning Representations (ICLR)*, 2025.

OpenAI. GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025), Datasets and Benchmarks Track*, 2025. arXiv:2507.02833; introduces the IFBench benchmark.

Kenneth F. Schulz, Douglas G. Altman, and David Moher. CONSORT 2010 statement: Updated guidelines for reporting parallel group randomised trials. *BMJ*, 340:c332, 2010.

Herbert A. Simon and Joseph B. Kadane. Optimal problem-solving search: All-or-none solutions. *Artificial Intelligence*, 6(3):235–247, 1975.

Ankur Sinha and Tanmay Khandait. Impact of news on the commodity market: Dataset and results. In *Advances in Information and Communication (FICC)*, 2021.

Jiayuan Su, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. CP-Router: An uncertainty-aware router between LLM and LRM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33065–33073, 2026.

Antonios Valkanas, Soumyasundar Pal, Pavel Rumiantsev, Yingxue Zhang, and Mark Coates. C3PO: Optimized large language model cascades with probabilistic cost constraints for reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. HelpSteer2-Preference: Complementing ratings with preferences. In *International Conference on Learning Representations (ICLR)*, 2025.

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

An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Alex Young et al. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*, 2024.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-R1: Teaching LLMs multi-round routing and aggregation via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 141233–141265, 2025a.

Xuechen Zhang, Zijian Huang, Ege Onur Taga, Carlee Joe-Wong, Samet Oymak, and Jiasi Chen. Efficient contextual LLM cascades through budget-constrained policy learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91691–91722, 2024.

Yiqun Zhang, Hao Li, Jianhao Chen, Hangfan Zhang, Peng Ye, Lei Bai, and Shuyue Hu. Beyond GPT-5: Making LLMs cheaper and better via performance-efficiency optimized routing. In *Proceedings of the 7th International Conference on Distributed Artificial Intelligence (DAI)*, pp. 122–129, 2025b.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. miniF2F: a cross-system benchmark for formal Olympiad-level mathematics. In *International Conference on Learning Representations (ICLR)*, 2022.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A DEFINITIONS, POLICY CLASSES, AND SCOPE OF THE GUARANTEES

This appendix fixes the objects that Sections 2 and 3 use, states the recursion that Algorithm 1 instantiates, and records, for every guarantee in the paper, which object it concerns. Every symbol is used with one meaning throughout the hand-written text; the generated tables and figures carry the policy's internal name, *StateCompose*, and use the word *certificate* both for the run-time verdict and for the cell-level gates defined in Section A.5.

A.1 REQUESTS, ANSWERS, VERIFIERS, AND CERTIFICATES

Definition 3 (Request, answer, verifier, certificate)**.** A benchmark role holds requests $x_i \in \mathcal{X}$, $i \in \{1, \dots, n\}$. An answer is a string $y \in \mathcal{Y}$. Correctness is the benchmark's official verdict $u_x : \mathcal{Y} \to \{0, 1\}$, computed from a hidden reference that no policy reads. A *public verifier* is a map $v_x : \mathcal{Y} \to \{0, 1\}$ computable from the request and the answer alone; $v_x(y) = 1$ is a *certificate* for y. The verifier is *exact* for the checked property when $v_x(y) = 1$ implies $u_x(y) = 1$ for every $y \in \mathcal{Y}$.

Two facts about certificates hold throughout. A rejection, $v_x(y) = 0$, is not evidence that y is wrong; it only withholds the certificate. Acceptance certifies the checked property and nothing beyond it (Table 1). The paper uses two classes of exact certificate. In class (a) the verifier is sound for the scored outcome but not identical to it: a Lean kernel accepts a proof term, and the task is scored by the same kernel on the same statement; a schedule validator accepts a feasible schedule. In class (b) the verifier *is* the scored metric: the official Multi-IF and IFBench checkers and the official JSON-schema compliance function score the benchmark, and the same function, applied to the cheap model's answer, is the certificate. In class (b) a delivered certified answer scores a pass by construction, so a harm can arise only on the rejection path, when the large model is truncated under the residual budget. Public example tests for code are not exact for hidden-test correctness and are treated by Proposition 5.

A.2 ACTIONS, CAPS, REALIZED COSTS, AND THE COST CONTRACT

Definition 4 (Registered actions and costs)**.** A registered pool is a finite set of actions $\mathcal{A}$, $|\mathcal{A}| = m$, each a call to one model under a fixed prompt, sampler and generation cap. Action a has parameter count N_a, prompt length $I_a(x)$ on request x, and generation cap M_a. Its *complete cost cap* is $C_a(x) = N_a \cdot (I_a(x) + M_a)$ parameter-weighted tokens (PWT) for an open checkpoint, and the released input and output prices applied to $I_a(x)$ and M_a (USD) for an API model. Its *realized cost* $c_a(x) \in [0, C_a(x)]$ is the same accounting applied to the tokens actually processed. One action $P \in \mathcal{A}$ is the *large model*: the strongest registered model in its registered configuration, whose answer is delivered whenever nothing better is certified.

Definition 5 (Cost contract)**.** The per-request budget is $B(x) = C_P(x)$, the large model's complete cap. A policy satisfies the *per-request clause* when every call it makes is admitted only if the call's complete cap fits the remaining budget, so that the cumulative realized cost never exceeds $B(x)$ on any request. A decision that would exceed the cap is replaced by the large model's answer and counted as a *violation*. A policy satisfies the *aggregate clause* on a role when the sum of its realized costs over the role's requests, and over each of the two mechanism halves separately, is at most θ times the sum of the large model's realized costs, with $\theta = 0.70$ preregistered.

The cost unit is model inference only: the verifier's own cost is excluded for both arms, and neither unit measures latency. For the finite-library cells (IFEval-es, M-IFEval and RCPSP) a separate analytic FLOP audit is applied to the same frozen paths; there the contract is recorded on four coordinates, the maximum-request and the average ratio under each of the PWT and FLOP ledgers. PWT-only admission does not guarantee the FLOP cap; the FLOP audit certifies only the recorded paths it checks.

A.3 PUBLIC STATE, POLICIES, AND THE RECURSION

Definition 6 (Public state and public context)**.** The *public state* $s \in \mathcal{S}$ of a request is everything a policy may read: the released context $\phi(x) \in \Phi$ (benchmark identity and its released subtype), the answers produced so far, and their certificates. No label, judge score or model confidence is part of s.

Table 1: Application scope is determined by the checked property. Acceptance does not certify any property other than the checked one.

Request type	Publicly checkable property	Not implied by acceptance
Constrained text	Counts, required phrases, format	Truth or general helpfulness
Planning / formal proof	Feasibility / proof-checker acceptance	Optimality / an unstated theorem
Code or SQL	Passing the supplied tests or fixtures	All-input or hidden-test correctness

A policy is a map $\pi : \mathcal{S} \times \mathbb{R}_{\geq 0} \to \mathcal{A} \cup \{\text{stop}\}$ from the public state and the remaining budget b to the next call or to stopping. Its value $V : \mathcal{S} \times \mathbb{R}_{\geq 0} \to [0, 1]$ satisfies the recursion

$$V(s, b) = \max \Big\{ U(s, b),\ \max_{a \in \mathcal{A}(s,b)} \mathbb{E}\big[V(S_a,\ b - c_a) \,\big|\, s\big] \Big\}, \qquad \mathcal{A}(s, b) = \{a \in \mathcal{A} :\ C_a \leq b\}, \quad (1)$$

where the expectation is over the call's answer, realized cost and next public state S_a: admission is on the complete cap, $\mathcal{A}(s, b) = \{a \in \mathcal{A} : C_a \leq b\}$, the transition charges the realized cost c_a, and the stopping utility $U(s, b)$ equals 1 on a state that records a certified answer and otherwise the accuracy of the large model's recorded sample delivered under the residual cap b (a truncated sample scores 0). When $\mathcal{A}(s, b)$ is empty the inner maximum is omitted and the policy stops.

Proposition 6 (Stopping on a certificate)**.** *Let the verifier be exact and let the public state s record a certified answer. Then $V(s, b) = U(s, b) = 1$ for every $b \geq 0$. Every policy that continues from (s, b) has expected score at most 1 and expected further cost at least 0, so stopping at (s, b) attains the largest expected score and the least further cost among all policies from (s, b).*

Proof. Exactness gives $u_x(y) = 1$ for the certified answer y, so $U(s, b) = 1$. Every score is at most 1, so every term of (1) is at most 1 and $V(s, b) \leq 1 = U(s, b) \leq V(s, b)$. A continuation calls at least one action, whose realized cost is nonnegative, and its expected score is one term of the inner maximum, hence at most 1. □

This is the only optimality that the recursion establishes. Which certifiers to call before the first certificate, and in what order, is chosen on the fit role from a finite library, and is cost-optimal only under the assumptions of Remark 7. The two families implemented in this paper are the following special cases.

Definition 7 (Certificate stopping)**.** The registered actions are certifiers $a_1, \ldots, a_k$ in a registered order and the large model P. The policy calls a_j when $C_{a_j} \leq b$, charges c_{a_j}, stops and delivers y_{a_j} if $v_x(y_{a_j}) = 1$, and after the last rejection delivers the large model's recorded sample replayed under the residual cap b.

Definition 8 (Root rule and agreement cascade)**.** For a request with no exact verifier the state is the released context $\phi(x)$. A root rule assigns to each context either the large model, a single registered action A delivered when $C_A \leq B(x)$, or an agreement cascade $\text{agree}[A|B]{>}P$ that calls A and B when both caps fit, delivers y_A when the two answers agree under the context's answer normalization, and otherwise delivers the large model's sample under the residual budget.

Algorithms 1 and 2 state the two families. Every result in this paper is a sealed replay: each action's outputs were generated once under its registered configuration, recorded with their realized costs, and the policy is evaluated by replaying the recorded outputs in the registered order. The paper does not claim online realizability.

A.4 Data roles and information access

Definition 9 (Roles)**.** Every benchmark is split by a fixed public function of the request text (or of its semantic parent, so that translations and paraphrases share a role) into four outcome-disjoint roles $r \in \{\text{fit}, \text{mech}, \text{val}, \text{conf}\}$. The fit role may read outcomes and selects the policy from a declared finite candidate library: 41 certifier orders for IFEval-es; one registered certifier for the 32B cells; one action per released context from up to 19 alternatives over 21 contexts on the routing pools. The finite-library cells carry a separate FLOP audit of the same frozen paths. The mechanism role is split into two deterministic halves $h \in \{1, 2\}$ and gates the frozen policy. The validation role is opened

Algorithm 1 Certificate stopping on a request x with an exact verifier v_x, certifiers $a_1, \ldots, a_k$ in registered order, large model P and caps C. The delivered answer costs $c_\pi(x) \le C_P(x)$.

$b \leftarrow C_P(x)$ ▷ remaining budget starts at $B(x)$
for $i = 1, \ldots, k$ **do**
  if $C_{a_i}(x) \le b$ **then** ▷ run a_i only if its cap fits
    $y_i \leftarrow$ answer of a_i on x ▷ costs $c_{a_i}(x) \le C_{a_i}(x)$
    $b \leftarrow b - c_{a_i}(x)$ ▷ charge the realized cost
    if $v_x(y_i) = 1$ **then return** y_i ▷ certified: stop
return P's recorded answer truncated at cost b ▷ scores 0 if truncated

Algorithm 2 Root rule for one request x with no exact verifier and budget $B(x) = C_P(x)$. Every branch is frozen before a closed role is opened; the only run-time signals are agreement and released fields.

Require: released context $\phi(x)$; registered actions with complete caps C_a; large model P
$b \leftarrow B(x)$
$\pi \leftarrow$ the action registered for $\phi(x)$: a single action A, a cascade $\mathrm{agree}[A|B]{>}P$, or P if none is registered
if $\pi = A$ and $C_A \le b$ **then return** call A
else if $\pi = \mathrm{agree}[A|B]{>}P$ and $C_A + C_B \le b$ **then** call A and B; $b \leftarrow b - c_A - c_B$; **return** y_A if the answers agree under the context's normalization, else P's recorded sample replayed under b
return the large model's answer at cost c_P (an over-cap decision is a violation)

once for a screening decision. The confirmation role is untouched by any model output before an authorization that names the frozen policy, and its outcomes are the only numbers the paper calls results.

An authorization is a written declaration, fixed before a role is opened, that names the models, the prompts, the caps, the sampler, the verifier, the role assignment and the gates. A confirmation role is opened at most once for a given registered policy and comparator; a failed gate is recorded as a frozen negative and the large model alone is replayed byte for byte. The fitted objects are the certifier order and caps of a certificate-stopping cell (a library of 41 orders for the IFEval-es cell, a single registered certifier for the 32B-comparator cells) and the registered action per context of a root rule (up to 19 alternatives per context over 21 contexts on the official routing pools). No router is trained by the method.

A.5 Signed advantage, exact bound, and gates

Definition 10 (Signed advantage and exact lattice bound)**.** On a role, the signed advantage of the policy over P on request i is $\Delta_i = u_{x_i}(y_\pi) - u_{x_i}(y_P) \in D = \{-1, 0, +1\}$ (harm, tie, rescue). With n_+ rescues and n_- harms among n requests and confidence level $1 - \delta$, the exact lattice lower bound on $\mathbb{E}[\Delta]$ is $\ell = \mathrm{CP}^-_{\delta/3}(n_+, n) - \mathrm{CP}^+_{\delta/3}(n_-, n)$, where, for k successes in n trials and $B(n, p)$ a binomial random variable with n trials and success probability p, the one-sided Clopper–Pearson bounds (Clopper & Pearson, 1934; Miller, 2024) at level $\eta \in (0, 1)$ are

$$\mathrm{CP}^-_\eta(k, n) \triangleq \sup\{p \in [0, 1] : \Pr[B(n, p) \ge k] \le \eta\},$$

and

$$\mathrm{CP}^+_\eta(k, n) \triangleq \inf\{p \in [0, 1] : \Pr[B(n, p) \le k] \le \eta\},$$

with $\mathrm{CP}^-_\eta(0, n) = 0$ and $\mathrm{CP}^+_\eta(n, n) = 1$; equivalently, $\mathrm{CP}^-_\eta(k, n)$ is the η-quantile of the $\mathrm{Beta}(k, n - k + 1)$ distribution and $\mathrm{CP}^+_\eta(k, n)$ the $(1 - \eta)$-quantile of $\mathrm{Beta}(k + 1, n - k)$. Each bound holds with probability at least $1 - \eta$, so ℓ holds with probability at least $1 - 2\delta/3 \ge 1 - \delta$, and δ is split equally over the declared support D.

Definition 11 (Gates)**.** A cell passes the *efficiency gate* when its cost ratio is at most $\theta = 0.70$ on the role and on both mechanism halves, its exact 95% lower bound ℓ is at least -1 point, and it has zero violations. It passes the *improvement gate* when both halves are positive, $\ell > 0$, the relative error

reduction over the strongest fixed action is at least 10%, and the exact McNemar test, a binomial test of the n_+ rescues against the n_- harms, is significant after Holm correction within the declared family. Both gates are declared in the mechanism contract before the confirmation role opens and are not relaxed afterwards.

A cell is *confirmed* when its confirmation role was opened once for the frozen policy and it passed the gate that its own authorization declared. For the cells sealed before the efficiency gate existed (IFEval-es, M-IFEval, the 128-task MiniF2F cell, HEADLINES, SNLI and the MBPP root), that gate is the one named in Appendix E; three of them (HEADLINES, M-IFEval, SNLI) sit above $0.70\times$ and would not pass the efficiency gate. A *development* row is a mechanism or validation result; a *frozen negative* is a cell whose gate failed; a *diagnostic* is a post-label audit. The generated tables label every row with one of these words.

A.6 Scope of each guarantee

Table 2 records, for each statement proved in this paper, the object it concerns. None of the statements is a claim about an unrestricted population of requests or an unrestricted policy class.

Routing contexts scored by a public verifier. The statements of Table 2 that require an exact certificate apply only where a public verifier is registered. For each of the 21 contexts of Table 16 we read the official scoring function of the pinned LLMRouterBench release. We ask whether that function reads a record field that the prompt does not display, and whether it calls a grader model. When the function reads no such field and calls no grader model, the official verdict is computable from the request and the answer, and the verdict is then itself a public verifier of class (b). Performance-20's MBPP is the only context that meets this condition. Its verdict executes a program formed from the answer and the assertions that the prompt displays, under a 10-second limit. A new execution therefore reproduces a recorded verdict only in an equivalent execution environment, and the record does not measure that agreement. No Performance-Cost context meets the condition. The MMLU-Pro, GPQA, BBH, WinoGrande, MATH-500, AIME, LiveMathBench and FinQA verdicts compare the answer with a reference that the prompt does not display. The LiveCodeBench verdict runs test cases that the prompt does not display. The HLE and SimpleQA verdicts call a grader model that reads a reference that the prompt does not display. The ArenaHard verdicts call a grader model that compares the answer with a baseline answer. For every context except Performance-20's MBPP, the analysis shows only that the official verdict reads a field that the prompt does not display or calls a grader model. It does not search for a different exact public verifier.

A post hoc string comparison checks the stored programs on the fit and mechanism roles of Performance-20's MBPP; it executes no program and cannot change any verdict. Of the 9,520 recorded outputs on those roles, 9,497 store a program that ends with the assertions displayed in exactly one released MBPP prompt. The 23 recorded outputs that do not end in this way each have 2,000 characters, the largest length of a stored program on those roles, and contain no released assertion. The root rule of Performance-20's MBPP delivers the answer of Qwen2.5-Coder-7B-Instruct without calling the verifier, so the confirmed MBPP cell is a model-selection result. The paper evaluates certificate stopping on no routing context.

Outside the routing pools, the official checker is an exact public verifier of class (b) for IFEval, M-IFEval, Multi-IF, IFBench and JSONSchemaBench. An exact public verifier of class (a) exists for MiniF2F and ProofNet, where it is the Lean kernel, and for RCPSP, where it is a schedule validator. The official verdicts of HEADLINES, SNLI, LiveCodeBench and the EvalPlus release of MBPP depend on a reference or on test cases that the request does not contain. The routing-pool classifications are read from scoring code. The classifications outside the routing pools follow the checker descriptions of Appendix E.

Pretraining exposure. The roles of Section 3 protect every reported number against our own adaptive selection, and they do not protect it against pretraining exposure. Pretraining exposure of a benchmark is what the literature calls contamination of that benchmark. The role of a request is a function of the request, or of its semantic parent, through a fixed public function of the request text, so the split is neither a temporal split nor a provenance split. Whether a model saw a request or its reference during pretraining is fixed before any role is opened, and nothing in this paper

measures it. If that split is independent of exposure, the confirmation role contains exposed requests in the same expected proportion as the fit role. Every sealed result therefore describes a released benchmark together with its exposure, and no result describes requests absent from pretraining data. Every benchmark in the record is a public release, and nothing in the record establishes that the requests or references of a benchmark postdate the training data of the models evaluated on it. The confirmed rows are measured on IFEval, M-IFEval, Multi-IF, IFBench, JSONSchemaBench, MiniF2F, HEADLINES, SimpleQA and the LLMRouterBench release of MBPP. The SNLI row, confirmed under its own earlier gate, is measured on SNLI. The development rows of Appendix E are measured on the public releases named there.

Exposure acts on the cost ratio and on the signed accuracy gain in different ways. By Theorem 1, the cost ratio of one-certifier stopping equals $\alpha+(1-q_a)\cdot\kappa\cdot\tau$, and its derivative in the certificate rate q_a is $-\kappa\cdot\tau\leq 0$. Suppose that exposure of the certifier raises q_a and leaves α, κ and τ unchanged. Then the cost ratio on exposed requests is smaller than the cost ratio on unexposed requests. Exposure can therefore make a reported cost ratio smaller than its value on requests absent from pretraining data. The rate q_a is computed from the verdicts of the verifier on the answers of the certifier. The factor α is computed from the realized costs of the certifier and of the large model. The factors κ and τ are computed from the realized costs of the large model and from its costs under the residual cap. No factor reads the hidden reference, so the cost ratio on a new set of requests can be measured without labels.

An exact verifier also bounds harm without labels. If $v_x(y_a)=1$, then $u_x(y_a)=1$ whatever process produced y_a, memorization included, so $\Delta_i\geq 0$ on every certified request. On a rejected request the policy delivers the large model's recorded sample under the residual cap, and a truncated sample scores 0 (Appendix A.3). The delivered answer on a rejected request therefore scores differently from y_P only when the residual cap truncates y_P. The number of harms n_- is at most the number of rejected requests on which the residual cap truncates the large model, and that number needs no label.

The signed accuracy gain has no label-free bound of this kind. Under one-certifier stopping with an exact verifier, the delivered answer on a rejected request scores at most $u_x(y_P)$, so a rescue occurs only on a certified request and $\mathbb{E}[\Delta]=\Pr[v_x(y_a)=1,\,u_x(y_P)=0]-\Pr[\Delta=-1]$. The rescue term $\Pr[v_x(y_a)=1,\,u_x(y_P)=0]$ reads the hidden reference through $u_x(y_P)$. Exposure of the certifier can raise the rescue term by raising q_a. Exposure of the large model can lower the rescue term by lowering the error rate of the large model. The record therefore determines neither the sign nor the size of the gain on requests absent from pretraining data. The confirmed rows of HEADLINES, SimpleQA and the LLMRouterBench release of MBPP and the SNLI row deliver answers that carry no certificate. On those rows the gain is a difference between two accuracies measured against the hidden reference. A difference between the exposures of the models involved can move that gain in either direction.

A.7 Correspondence with the finite-library cells

The IFEval-es, M-IFEval and RCPSP cells and the IFBench diagnostic were produced by an earlier implementation whose vocabulary differs: its *anchor* a_0 is the first certifier of Definition 7; its *donors* are the further certifiers; its *comparator* L is the large model P; its *residual-budget recycling* is certificate stopping with the large model's cap as the budget and earliest-certificate delivery; and its 41-order class is the finite library from which the fit role selected the certifier order. Two differences are substantive and are kept visible in the record. First, those cells admit actions by PWT only, with a separate FLOP audit on the same frozen paths (the IFBench diagnostic used a dual-ledger admission). Second, the 32B- and 72B-comparator cells use a single registered certifier and no order search, so the only fitted objects there are the caps.

B Proofs of the results of Section 2

Throughout, X is the random request of Definition 1, $\mathbb{E}_n$ and $\Pr_n$ are its averages over the n requests of one role, and $\mathbb{E}$ and $\Pr$ are the expectation and the probability under a general request distribution D. Appendix C states and proves the general versions of these results, with several certifiers; the proofs below use nothing from it.

Table 2: Scope of the guarantees. Optimizer: exact over a stated class, or finite library. Cost: cap or realized. Mode: expected or deterministic. Sample: finite role or population. Certificate: exact required or imperfect allowed. Policy: fixed, or selected from data.

Statement	Optimizer	Cost	Mode	Sample	Certificate	Policy
Thm. 1 cost identity	none (identity)	realized	deterministic on the role	finite role	not needed	fixed or selected
Lemma 3 selection factor	none	realized	population conditional expectation	population; monotone latent difficulty	not needed	fixed
Remark 7 index order	finite library of orders	cap, constant	expected	population rates	exact	fixed
Thm. 4 admission	one added certifier	realized	expected	population, conditioned on reaching the certifier	exact	fixed
Prop. 5 harm bound	none	none	expected rates	population; finite-sample corollary	imperfect	fixed
Prop. 12 fragmentation	none	none	deterministic (property of the bound)	finite role	lattice bound on Δ	fixed cell
Prop. 10 simultaneous validity	any selection rule	cap (admissibility)	probability $1-\delta$	finite role, population averages	lattice bound on Δ	data-selected
Prop. 11 dominance	deployed tree	cap	(i) deterministic, (ii) probability $1-\delta$	fit distribution	lattice bound on Δ	data-selected

B.1 Proof of Theorem 1 (cost identity)

Proof. Let one certifier a run on every request, so that $C_a(x) \le B(x)$ for every x. By Algorithm 1, on a request x the policy pays $c_a(x)$ for the certifier and then, only if $x \in R_a$, pays $c'_P(x)$ for the large model truncated at the remaining budget $B(x) - c_a(x)$. Hence its realized cost is

$$c_\pi(x) = c_a(x) + \mathbf{1}_{R_a}(x) \cdot c'_P(x) \qquad \text{for every } x \in \mathcal{X}.$$

Averaging over the n requests and dividing by $\mathbb{E}_n[c_P(X)] > 0$, which is positive because $\Pr_n[R_a] > 0$ and $\mathbb{E}_n[c_P(X) \mid R_a] > 0$,

$$\Gamma_n = \frac{\mathbb{E}_n[c_\pi(X)]}{\mathbb{E}_n[c_P(X)]} = \frac{\mathbb{E}_n[c_a(X)]}{\mathbb{E}_n[c_P(X)]} + \frac{\mathbb{E}_n[\mathbf{1}_{R_a}(X) \cdot c'_P(X)]}{\mathbb{E}_n[c_P(X)]} = \alpha + \frac{\Pr_n[R_a] \cdot \mathbb{E}_n[c'_P(X) \mid R_a]}{\mathbb{E}_n[c_P(X)]},$$

where the last step is the definition $\mathbb{E}_n[f(X) \mid R_a] = \mathbb{E}_n[\mathbf{1}_{R_a}(X) \cdot f(X)] / \Pr_n[R_a]$ of Definition 1. Since $\Pr_n[R_a] = 1 - q_a$ and $\mathbb{E}_n[c_P(X) \mid R_a] > 0$,

$$\frac{\Pr_n[R_a] \cdot \mathbb{E}_n[c'_P(X) \mid R_a]}{\mathbb{E}_n[c_P(X)]} = (1 - q_a) \cdot \frac{\mathbb{E}_n[c_P(X) \mid R_a]}{\mathbb{E}_n[c_P(X)]} \cdot \frac{\mathbb{E}_n[c'_P(X) \mid R_a]}{\mathbb{E}_n[c_P(X) \mid R_a]} = (1 - q_a) \cdot \kappa \cdot \tau.$$

Finally $\tau \le 1$ because $c'_P(x) \le c_P(x)$ for every x: the truncated call replays the large model's recorded answer, and a shorter generation cap can only shorten it. The proof uses only the acceptance events, not the correctness of accepted answers, so it does not need the verifier to be exact. □

B.2 Proof of Proposition 2 (prediction under proportional cost)

Proof. The rejection event $R_a = \{x \in \mathcal{X} : v_x(y_a(x)) = 0\}$ is determined by the certifier's answer and the verifier, so it is the same for P and for P', and so is $q_a = 1 - \Pr_n[R_a]$. Applying Theorem 1 to the policy with P' in place of P gives $\Gamma'_n = \alpha' + (1 - q_a) \cdot \kappa' \cdot \tau'$ with

$$\alpha' = \frac{\mathbb{E}_n[c_a(X)]}{\mathbb{E}_n[c_{P'}(X)]} = \frac{\mathbb{E}_n[c_a(X)]}{\lambda \cdot \mathbb{E}_n[c_P(X)]} = \frac{\alpha}{\lambda}$$

and

$$\kappa' = \frac{\mathbb{E}_n[c_{P'}(X) \mid R_a]}{\mathbb{E}_n[c_{P'}(X)]} = \frac{\lambda \cdot \mathbb{E}_n[c_P(X) \mid R_a]}{\lambda \cdot \mathbb{E}_n[c_P(X)]} = \kappa,$$

and, using $c'_{P'}(x) = \lambda \cdot c'_P(x)$ on R_a,

$$\tau' = \frac{\mathbb{E}_n[c'_{P'}(X) \mid R_a]}{\mathbb{E}_n[c_{P'}(X) \mid R_a]} = \frac{\lambda \cdot \mathbb{E}_n[c'_P(X) \mid R_a]}{\lambda \cdot \mathbb{E}_n[c_P(X) \mid R_a]} = \tau.$$

The hypotheses of Theorem 1 hold for P' because $\mathbb{E}_n[c_{P'}(X) \mid R_a] = \lambda \cdot \mathbb{E}_n[c_P(X) \mid R_a] > 0$. The second hypothesis is not implied by the first. The remaining budget of P' on a rejected request is $C_{P'}(x) - c_a(x) = \lambda \cdot C_P(x) - c_a(x)$, which is not λ times the remaining budget $C_P(x) - c_a(x)$ of P unless $c_a(x) = 0$; so the two models can be truncated at different points of the same generation, and the hypothesis $c'_{P'}(x) = \lambda \cdot c'_P(x)$ states that they are not. □

B.3 Proof of Lemma 3 (selection under monotone difficulty)

Proof. Write $\mathbf{1}_{R_a}$ for $\mathbf{1}_{R_a}(X)$ and c_P for $c_P(X)$. Since $\mathbf{1}_{R_a}$ is an indicator with $\Pr[R_a] > 0$,

$$\mathrm{Cov}(\mathbf{1}_{R_a}, c_P) = \mathbb{E}[\mathbf{1}_{R_a} c_P] - \Pr[R_a] \cdot \mathbb{E}[c_P] = \Pr[R_a] \cdot \big(\mathbb{E}[c_P \mid R_a] - \mathbb{E}[c_P]\big),$$

so $\kappa = \mathbb{E}[c_P \mid R_a]/\mathbb{E}[c_P] \geq 1$ if and only if $\mathrm{Cov}(\mathbf{1}_{R_a}, c_P) \geq 0$. The law of total covariance splits this covariance into a part within difficulties and a part across them:

$$\mathrm{Cov}(\mathbf{1}_{R_a}, c_P) = \mathbb{E}\big[\mathrm{Cov}(\mathbf{1}_{R_a}, c_P \mid Z)\big] + \mathrm{Cov}\big(r(Z), m(Z)\big),$$

because $\mathbb{E}[\mathbf{1}_{R_a} \mid Z] = r(Z)$ and $\mathbb{E}[c_P \mid Z] = m(Z)$. The first term is nonnegative by hypothesis. For the second, let Z' be an independent copy of Z. By comonotonicity, $(r(Z) - r(Z')) \cdot (m(Z) - m(Z')) \geq 0$ pointwise, and expanding the product gives

$$\mathrm{Cov}\big(r(Z), m(Z)\big) = \tfrac{1}{2} \cdot \mathbb{E}\big[(r(Z) - r(Z')) \cdot (m(Z) - m(Z'))\big] \geq 0.$$

Hence $\mathrm{Cov}(\mathbf{1}_{R_a}, c_P) \geq 0$ and $\kappa \geq 1$. If $\kappa = 1$, both terms vanish, so the nonnegative product $(r(Z) - r(Z')) \cdot (m(Z) - m(Z'))$ is zero almost surely. Suppose that neither $r(Z)$ nor $m(Z)$ is almost surely constant: there are values z_1, z_2 of positive probability with $r(z_1) < r(z_2)$ and values z_3, z_4 of positive probability with $m(z_3) < m(z_4)$. A zero product on the pair (z_1, z_2) forces $m(z_1) = m(z_2) =: \mu$, and one of $m(z_3), m(z_4)$ differs from μ, say $m(z_3)$; zero products on the pairs (z_1, z_3) and (z_2, z_3) then force $r(z_1) = r(z_3) = r(z_2)$, a contradiction. Hence $r(Z)$ or $m(Z)$ is almost surely constant. Nothing in the argument uses an order on $\mathcal{Z}$. □

B.4 Proof of Theorem 4 (admission)

Proof. All averages are over the requests that reach a. Write $\bar{c} \triangleq \mathbb{E}[c_a(X)]$, $c_T \triangleq \mathbb{E}[c_P(X; b(X))]$ and $c_T^- \triangleq \mathbb{E}[c_P(X; b(X) - c_a(X)) \mid R_a]$, so that $\rho = \bar{c}/c_T$ and $e = (c_T - c_T^-)/\bar{c}$. Without a, the expected cost of the large model is c_T. With a, the policy pays $c_a(x)$ on every reaching request and, on the requests in R_a, pays $c_P(x; b(x) - c_a(x))$ for the large model; by the definition of a conditional average, its expected cost is

$$\bar{c} + \mathbb{E}\big[\mathbf{1}_{R_a}(X) \cdot c_P(X; b(X) - c_a(X))\big] = \bar{c} + \Pr[R_a] \cdot c_T^- = \bar{c} + (1 - q_a) \cdot (c_T - e \cdot \bar{c}).$$

The difference between the two policies is therefore

$$\Delta = \bar{c} + (1 - q_a) \cdot (c_T - e \cdot \bar{c}) - c_T = \bar{c} \cdot \big[1 - e \cdot (1 - q_a)\big] - q_a \cdot c_T.$$

Dividing by $c_T > 0$, $\Delta < 0$ if and only if $\rho \cdot [1 - e \cdot (1 - q_a)] < q_a$, that is, $\rho \cdot (1 - e) < q_a \cdot (1 - \rho \cdot e)$, which for $1 - \rho \cdot e > 0$ is $q_a > \rho \cdot (1 - e)/(1 - \rho \cdot e)$. At $e = 0$ the right-hand side is ρ. No assumption on the distribution of $c_a(X)$, $b(X)$ or the rejection event is used: the statement is an identity between the averages that define q_a, ρ and e. □

B.5 Proof of Proposition 5 (harm of an inexact verifier)

Proof. With $U_P \triangleq u_X(y_P(X))$ the correctness of the large model's answer, a harm is the event $\{V_a = 1, U_P = 1, U_a = 0\}$. It is contained in $\{V_a = 1, U_a = 0\}$, whose probability is $\Pr[V_a = 1] \cdot \Pr[U_a = 0 \mid V_a = 1] = q_a \cdot (1 - p_a)$. For an exact certificate, $v_x \leq u_x$ for every request gives $U_a = 1$ whenever $V_a = 1$, so $p_a = 1$ and the bound is zero. □

C Certificate laws on the sealed cells and portfolio design

This appendix reads the cost identity of Section 2 on each sealed cell, states the classical index order, and develops admission under a shared budget with the knapsack theorem; the proofs of the main-text results are in Appendix B. A *certificate action* $a \in \mathcal{A}$ is a model call with complete cost cap C_a,

realized cost $c_a \le C_a$, and a public verifier v_x that accepts its delivered answer with probability q_a (its *certificate rate*); acceptance implies correctness when the certificate is exact. The large model P has cap C_P, realized cost c_P when run alone, and accuracy σ_P. A certificate-first portfolio runs $a_1, \ldots, a_m$ in order, delivers at the first acceptance, and otherwise runs P under the residual budget. An action is *reached* when its call is made: no earlier action accepted and its complete cap fits the residual budget. Expectations are averages over the requests of one role unless a population is named. The cost ratio of a cell is the policy's average realized cost divided by the large model's average realized cost when run alone.

Lemma 3 of the main text signs the selection factor. Its hypothesis is one-sided, and that is what makes it usable. It does not ask that rejection and cost be unrelated once difficulty is fixed, only that the relation, if any, is not negative; a direct positive dependence, as when the certifier rejects because the large model was itself truncated, adds to the first term and strengthens the conclusion. What the statement supplies is the second term, which no measurement of the two quantities at a fixed difficulty can provide: comonotone dependence on a common latent cause signs the covariance by itself. The sealed cells agree with the conclusion: on the theorem-proving cells of Table 3, κ ranges from 1.10 to 1.74. The statement is about κ alone; the prefix trim $\tau \le 1$ acts in the opposite direction, so the product $\kappa\tau$ can fall below one on a cell where the residual cap truncates the large model heavily, as it does on the IFBench classic-constraint cell.

Under the empirical measure of a role the identity is arithmetic over logged costs, so it holds to rounding on every sealed cell by construction. Table 3 evaluates the three factors on every sealed Lean cell. On MiniF2F the certifier's share α is 0.03–0.04 of the comparator's realized cost, the prefix trim τ is 0.96–0.97, and the selection factor κ is 1.3–1.7: the 7B prover fails on exactly the statements on which the 32B prover runs toward its cap, which is why the realized cost ratios (0.53 on the mechanism role, 0.69 on MiniF2F-test) sit well above the naive cap bound $C_{a_1}/C_P + (1 - q_1)$. On ProofNet the same identity explains the failed efficiency gate: the certificate rate drops to $q_1 = 0.18$, the comparator runs toward its cap on almost every statement ($\kappa = 1.10$), and the cost ratio is $0.02 + 0.82 \cdot 1.10 \cdot 0.96 = 0.89$. On the Multi-IF instruction-following cell the same three factors read $\alpha = 0.13$ (a 4B certifier against a 32B comparator), $q_1 = 0.77$, $\kappa = 1.37$ and $\tau = 0.94$ on the confirmation role, giving $0.13 + 0.23 \cdot 1.37 \cdot 0.94 = 0.43$, the realized cost ratio 0.432 (mechanism role: 0.14, 0.77, 1.33, 0.96, realized cost ratio 0.427). The turn series of the same cell is the law traced by one variable: with α, κ and τ almost fixed (0.12–0.13, 1.14–1.37, 0.94–0.95), the certificate rate falls from $q_1 = 0.77$ (turn 1) to 0.59 (turn 2) to 0.50 (turn 3) as the constraints accumulate, and the realized cost ratio rises from 0.43 to 0.58 to 0.68, crossing the 0.70 ceiling on one half at the third turn, which is where the preregistered gate stops the series. On the French and Spanish turn-1 cell the factors read $\alpha = 0.14$, $q_1 = 0.84$ and $\kappa\tau = 1.00$, giving $0.14 + 0.16 \cdot 1.00 = 0.30$, the realized cost ratio 0.297: a higher certificate rate than in the five-language cell and a comparator that does not run toward its cap on the rejected prompts; its second turn reads $\alpha = 0.13$, $q_1 = 0.68$ and $\kappa\tau = 1.10$, giving a cost ratio of 0.48, and its third turn $\alpha = 0.12$, $q_1 = 0.51$ and $\kappa\tau = 1.08$, giving 0.65: with q_1 and $\kappa\tau$ almost equal to the five-language third turn's, the cost ratio lands 0.02 under the ceiling instead of 0.02 over it, which is how thin the boundary is. On the IFBench classic-constraint cell the factors read $\alpha = 0.12$, $q_1 = 0.81$ and $\kappa\tau = 0.97$ (cost ratio 0.31); on its out-of-distribution subset they read $\alpha = 0.13$, $q_1 = 0.43$ and $\kappa\tau = 1.02$, which puts the cost ratio at 0.71 and fails the gate, the same law with the certificate rate again the moving factor. The structured-output cells (JSONSchemaBench) read $\alpha = 0.12$, $q_1 = 0.83$ and $\kappa\tau = 1.37$, giving a cost ratio of 0.36, and $\alpha = 0.12$, $q_1 = 0.89$ and $\kappa\tau = 1.36$, giving 0.27. The value of the identity is that it names the one factor a designer controls, q_1, and the one that the task distribution sets, κ. It is also predictive across comparators: with the certifier and its certificate rate fixed, replacing the 32B by the 72B rescales α by the parameter ratio $N_{32}/N_{72} = 0.45$ and leaves $\kappa\tau$ to the 72B's own behaviour on the rejected rows, so the cost ratio of each second-comparator cell was pre-registered from its 32B cell as $\alpha_{32} N_{32}/N_{72} + (1 - q_1)\kappa\tau_{32}$; the realized confirmation cost ratios (0.362, 0.212, 0.240, 0.289, 0.209 on the five turn-1 subsets of Table 8 whose 32B confirmation had been sealed) match the predictions (0.359, 0.223, 0.241, 0.287, 0.206) to 0.003 or better on four cells and to 0.011 on the French and Spanish cell, which says the 72B's selection factor on the certifier's rejects equals the 32B's to within noise. The sixth subset, IFBench's out-of-distribution subset, carries no confirmation-role prediction: the 32B series froze it at the mechanism stage, so the only pre-registered figure for it, 0.64, was computed on the disjoint mechanism rows and is reported as such rather than as a prediction of the confirmed cost ratio 0.633.

Table 3: The realized-cost identity of certificate stopping on every sealed Lean cell: $\text{ratio} = \alpha + (1 - q_1)\,\kappa\,\tau$ with α the certifier's share of the comparator's realized cost, q_1 its certificate rate, κ the comparator's conditional cost on the residual relative to its average, and τ the prefix trim. The naive cap bound $C_{a_1}/C_P + (1 - q_1)$ is shown for contrast; it is not a bound on the realized ratio because $\kappa > 1$: the certifier fails on exactly the tasks on which the comparator runs long.

cell	n	q_1	α	κ	τ	identity	realized	naive cap bound
MiniF2F-valid mechanism (v11m)	104	0.692	0.034	1.68	0.966	0.533	0.533	0.349
MiniF2F-test confirmation (v11c)	244	0.471	0.029	1.29	0.972	0.692	0.692	0.570
mechanism, seed 1 (v12)	104	0.750	0.031	1.74	0.972	0.453	0.453	0.291
mechanism, seed 2 (v12)	104	0.692	0.033	1.66	0.971	0.527	0.527	0.349
mechanism, seed 3 (v12)	104	0.673	0.034	1.63	0.973	0.553	0.553	0.368
ProofNet-valid mechanism (v13)	182	0.181	0.020	1.10	0.964	0.888	0.888	0.860

Remark 7 (Classical index order). For exact certifiers with constant costs $c_a = C_a$, jointly admissible on every request, whose acceptance events are mutually independent, every order that runs them until the first acceptance delivers the same accuracy, and the expected certifier cost is minimized by the ascending order of C_a/q_a, by an interchange of adjacent actions: running a then b costs $C_a + (1 - q_a) \cdot C_b$ against $C_b + (1 - q_b) \cdot C_a$, which is no larger if and only if $C_a/q_a \le C_b/q_b$. This is the least-cost testing sequence (Mitten, 1960) and optimal satisficing search (Simon & Kadane, 1975); we claim neither the statement nor the argument, and we state its assumptions because they are what a shared budget violates. Theorem 9 uses this order.

Both assumptions are stronger than the measured cells support: realized costs depend on the order, and the certifiers are not independent, since the joint rejection probability exceeds the product of the marginals on every cell. Dependence alone does not decide whether the index order is optimal; the selection factor $\kappa > 1$ of Theorem 1 is not evidence either way, because it measures the association between rejection and the large model's cost and not the association among certifiers. The registered order of every cell is therefore the fit-role choice among a finite library, optimal only under the assumptions above. The rule every confirmed cell instantiates is nevertheless the classical one: the first action is the cheapest certifier with a non-trivial certificate rate (a 7B code-first prover at 2.8% of the 32B cap; a 4B instruction model against a 32B comparator; a 235B non-thinking model at 0.4% of gpt-5's price), and the large model is last because its index is largest by definition.

C.1 Admission under a shared budget

Remark 7 settles *ordering* and says nothing about *admission*, which certifiers to run at all. In the classical setting that omission is harmless, because running a test does not change what the terminal decision is worth. Here it does: every certifier call shortens the large model's allowance, and the large model is truncatable. Write T for the terminal, the large model run under the residual budget b, with success probability $\sigma_T(b)$ and cost $c_T(b)$. The model of this subsection takes σ_T and c_T as functions of b alone (no selection effect of the certifiers on the terminal's requests), certifier costs c_j as constants, and acceptance events as independent. Writing a portfolio as a set A with rates q_j and costs c_j, the terminal is reached only when every member of A has rejected, and on that path every c_j has been paid, so

$$\mathrm{Acc}(A) \;=\; 1 - \bar{q}_A\big(1 - \sigma_T(B - c_A)\big), \qquad \bar{q}_A = \prod_{j \in A}(1 - q_j), \quad c_A = \sum_{j \in A} c_j. \tag{2}$$

Both $\bar{q}_A$ and c_A are symmetric, so *order does not affect accuracy at all*; it affects only expected cost, where Remark 7 applies. Admission is the separate question; Theorem 4 of the main text answers it for expected cost, and this subsection scores that answer on the sealed cells.

Table 4 scores both cost rules on the eleven cells that ran the two-stage and the three-stage policy on the same requests, six where the middle stage paid for itself and five where it did not. The classical index is right on nine, the elasticity-corrected condition on all eleven, and the two disagree on exactly the two cells where the classical one errs. The condition was derived after those cells were sealed, so eleven of eleven is a fit rather than a test. The cross-family ladder, whose cells were authorized before any comparator token existed, was registered in advance as the out-of-sample test of the condition; on that ladder the corrected condition is right on 24 of 24 cells and the classical index on 21 of 24, the two disagreeing on three cells (Table 19). This is one pre-registered rule

scored on one ladder, not a confirmation of the theory; an accuracy comparison is not discriminating there, since the three-stage policy raised accuracy on all 24 cells.

Remark 7 and Theorem 4 compare two portfolios that differ in one certifier. Neither says which portfolio is best. Under a budget that the certifiers and the large model share, the answer turns on one property of the large model: the rate at which its log-reliability grows with the budget it is given.

Write $\nu_i = -\log(1-q_i) > 0$ for the certifier weights, $\Lambda(b) = -\log\big(1-\sigma_T(b)\big)$ for the large model's log-reliability, assumed finite and nondecreasing on $[0,B]$, and $d_i = \nu_i/c_i$ for the density of certifier i. A *portfolio* is a set $S \subseteq \{1,\dots,m\}$ with $c(S) = \sum_{i\in S} c_i \le B$ together with an order in which to run it, and we abbreviate

$$H(S) \;=\; \sum_{i\in S} \nu_i + \Lambda\big(B - c(S)\big), \qquad H^* = \max\{H(S) : c(S) \le B\}. \tag{3}$$

The independent-acceptance model of Remark 7 gives $\Pr[\text{fail}] = \exp(-H(S))$. That model is an idealization, and the next statement says in which direction it errs, so that the design problem below is understood to optimize a surrogate rather than the realized failure probability.

Lemma 8 (The independent model is optimistic)**.** *Let Z be a latent difficulty. Suppose the rejection events of the certifiers and the failure event of the large model are conditionally independent given Z and each has a conditional probability nondecreasing in Z. Then the realized failure probability of the portfolio S is at least* $\exp(-H(S))$.

Proof. Conditionally on Z the events are independent, so the probability that all certifiers reject and the large model fails equals $\mathbb{E}_Z\big[\prod_{i\in S} r_i(Z)\, f(Z)\big]$ with each factor nondecreasing in Z. Harris's inequality (Harris, 1960) for nondecreasing functions of a single real random variable, applied inductively, bounds this below by $\prod_{i\in S} \mathbb{E}[r_i(Z)] \cdot \mathbb{E}[f(Z)]$, which is $\exp(-H(S))$. □

Lemma 8 strengthens the monotone-difficulty conditions of Lemma 3 by specifying conditional independence of the outcome events. A measured $\kappa \ge 1$ is consistent with selection but does not establish these stronger assumptions.

Theorem 9 (Portfolio design under a shared budget)**.** *Let $a_1,\dots,a_m$ be exact certifiers with deterministic costs $c_i > 0$ and acceptance probabilities $q_i \in (0,1)$. Assume that their acceptance events are mutually independent and independent of the large model's failure event at every remaining budget. Let the large model run after all rejections. Index the certifiers so that $d_1 \ge \cdots \ge d_m$, write $C_k = c_1 + \cdots + c_k$ and $S_k = \{a_1,\dots,a_k\}$, and let S_g be the best feasible prefix, the maximizer of $H(S_k)$ over $0 \le k \le m$ with $C_k \le B$.*

- *(i)* Order. *Among policies that run a set of certifiers until the first acceptance and then the large model, the failure probability depends on the set alone, and for each fixed feasible set the expected certifier cost is minimized by the ascending order of c_i/q_i. A shared budget therefore changes which certifiers to run, never the order in which to run them.*

- *(ii)* Affine log-reliability. *If $c(\{1,\dots,m\}) \le B$ and Λ is affine with slope λ on $[B - c(\{1,\dots,m\}), B]$, then $H(S) = \Lambda(B) + \sum_{i\in S}(\nu_i - \lambda c_i)$, so the design problem decouples and $\{i : \nu_i > \lambda c_i\}$ is optimal, with ties immaterial. At $\lambda = 0$ every exact certifier is admitted.*

- *(iii)* Greedy prefix. *Suppose that Λ is continuous. Let t^* maximize $g(t) + \Lambda(B-t)$ over $t \in [0, \min\{B, C_m\}]$, where g is the fractional-knapsack value. If $t^* = C_m$, all certifiers fit and S_m is optimal. Otherwise, let f be the index with $C_{f-1} \le t^* < C_f$, and let $\theta = (t^* - C_{f-1})/c_f \in [0,1)$. Then $H^* - H(S_g) \le \theta\,\nu_f$, so the failure probability of S_g is at most $(1-q_f)^{-\theta} \le (1-q_f)^{-1}$ times that of the optimal portfolio. Apart from attainment of the fractional maximum, only monotonicity is used. If Λ is in addition concave and $C_f \le B$, then $H^* - H(S_g) \le \min\big\{\theta\nu_f,\ \big((1-\theta)(c_f\lambda_f - \nu_f)\big)_+\big\}$ for any supergradient λ_f of Λ at $B - C_f$; in particular the prefix is optimal when $\lambda_f \le d_f$, and also, for any nondecreasing Λ, when all costs are equal.*

- *(iv)* Complexity. *Deciding whether some feasible portfolio attains a given failure probability is NP-complete, already for costs and budget given in binary, rational acceptance probabilities of $O(\log \sum_i w_i)$ bits, and the concave log-reliability $\Lambda(b) = \log(1+\lambda b)$. The*

hardness is weak: for positive integer costs and integer budget, a dynamic program uses $O(mB)$ exact arithmetic and comparison operations for any nondecreasing Λ supplied by an evaluation oracle. This is an operation count, not a bit-complexity claim for arbitrary real oracles. For the rational family above, the problem lies in P when the integer budget is polynomially bounded.

Proof. (i) Acceptance is absorbing, so on every history that has not stopped, every certifier run so far has rejected; the decision tree is a path and a policy is a set with an order. On that path the large model's budget is $B - c(S)$, so the failure probability is $\prod_{i \in S}(1 - q_i)\big(1 - \sigma_T(B - c(S))\big)$, which depends on S alone. Running the large model when the budget permits can only lower the failure probability, since $\sigma_T \geq 0$, so policies that stop without it are dominated. For the cost, exchanging adjacent positions j and $j+1$ changes the certifier cost by $\big(\prod_{i<j}(1-q_i)\big)\big(c_{j+1}q_j - c_j q_{j+1}\big)$, which is nonnegative exactly when $c_j/q_j \leq c_{j+1}/q_{j+1}$.

(ii) Affinity gives $\Lambda(B-c(S)) = \Lambda(B)-\lambda c(S)$ on the stated interval, and the feasibility hypothesis makes every subset admissible, so the exponent is maximized termwise.

(iii) If $t^* = C_m$, the relaxed maximizer is the feasible full set, which proves the endpoint claim. Otherwise the break index exists. Let $h^* = \sup\{\nu \cdot x + \Lambda(B - c \cdot x) : x \in [0,1]^m,\ c \cdot x \leq B\}$; the budget constraint cannot be dropped, since Λ is defined only on $[0, B]$ and the break index need not exist without it. Fix x in the relaxed feasible set with $t = c \cdot x < C_m$, and let k be the largest index with $C_k \leq t$. Then S_k is feasible, $\nu \cdot x \leq g(t) = \nu(S_k) + \theta(x)\nu_{k+1}$ by the exact form of the fractional knapsack value, and $\Lambda(B - t) \leq \Lambda(B - C_k)$ because Λ is nondecreasing. Adding the two gives $\nu \cdot x + \Lambda(B - t) \leq H(S_k) + \theta(x)\nu_{k+1} \leq H(S_g) + \theta(x)\nu_{k+1}$. Applying this at a maximizer of the relaxation yields the stated bound, since $H^* \leq h^*$. Continuity on the compact feasible set ensures attainment. The refinement under concavity follows by bounding the same gap with a supergradient of Λ at $B - C_f$. When $\lambda_f \leq d_f$, that bound vanishes. For equal costs, each fixed-cardinality subset is dominated in total weight by the density prefix of the same cardinality, with the same remaining budget. Maximizing over feasible cardinalities proves that special case.

(iv) Membership is immediate, since $\prod_{i \in S}(1 - q_i)$ is an exact rational and the comparison is by cross-multiplication. For hardness, reduce from Subset-Sum with weights $w_1, \ldots, w_m$ and target W. Put $T = \sum_i w_i$, $\varepsilon = 1/(4T^2)$, $c_i = w_i$, $B = W$, $q_i = \varepsilon w_i$, $\lambda = \varepsilon/2$ and $\vartheta = 1 - \varepsilon W + (\varepsilon W)^2/2$; all are rationals of $O(\log T)$ bits. Since $y \leq -\log(1 - y) \leq y + y^2/(2(1 - y))$, the total nonlinearity is at most $\frac{2}{3}\varepsilon^2 T^2 = \varepsilon/6$, so $\varepsilon w(S) \leq \sum_{i \in S} \nu_i \leq \varepsilon w(S) + \varepsilon/6$ for every S. A subset of weight exactly W leaves no budget and attains failure at most $e^{-\varepsilon W} \leq \vartheta$. Every feasible S of weight $s \leq W - 1$ has $-\log \Pr[\text{fail}] \leq \varepsilon W - (\varepsilon/2)(W - s) + \varepsilon/6 \leq \varepsilon W - \varepsilon/3$, and $\vartheta - e^{-\varepsilon W} \leq (\varepsilon W)^3/6 < e^{-\varepsilon W}(e^{\varepsilon/3} - 1)$, so no such S meets the target. The chain uses only $\lambda \leq \varepsilon/2$, so the same threshold certifies every rational λ in $[0, \varepsilon/2]$, including the concave family above. For the algorithm, $H(S)$ splits into a term in S and a term depending on S only through $c(S)$. For integer costs, let $A[i][t]$ be the largest total weight of a subset of the first i items with cost exactly t. Initialize $A[0][0] = 0$ and $A[0][t] = -\infty$ for $t > 0$. For $t \geq c_i$, the recurrence is $A[i][t] = \max\{A[i-1][t], A[i-1][t - c_i] + \nu_i\}$ with $\mathrm{OPT} = \max_{0 \leq t \leq B}(A[m][t] + \Lambda(B - t))$. For $t < c_i$, set $A[i][t] = A[i-1][t]$. There are $O(mB)$ operations and $B + 1$ oracle calls. In the rational family, one can instead store the minimum product of rejection probabilities at each exact cost and compare rational failure probabilities by cross-multiplication. Each product has polynomial bit length, establishing the stated P claim. □

The affine case is a sufficient condition for separable admission when all items fit. The concave example establishes weak NP-hardness for a particular family; it does not classify every non-affine reliability function. The dynamic program is pseudopolynomial in the integer budget. The measured policies use finite-library fitting, not this portfolio dynamic program. Finally, the approximation bound concerns the independent surrogate. Lemma 8 does not turn surrogate optimality into optimality for the dependent outcomes of the measured cells.

Proposition 5 of the main text bounds the harm of an inexact certificate. The bound covers false certificates only. On the rejection path a call to the large model replayed under the residual cap can also lose a request that the full cap would have won; those truncation harms are not bounded by Proposition 5 and are counted directly in every ledger. The two certificate classes in our experiments separate on this bound. Lean compilation is exact ($p_a = 1$): across the mechanism seed and three

Table 4: Admission of the middle certifier on the eleven cells that ran both the two-stage and the three-stage policy on the same requests. q_m is the middle stage's certificate rate on the requests it reaches, $\rho = c_m/c_T$ its cost relative to the terminal's, and e the measured truncation elasticity, the fraction of each unit the middle stage spends that the terminal then does not spend. The classical index admits when $q_m > \rho$; Proposition 4 admits when $q_m > \rho(1-e)/(1-\rho e)$, which reduces to the classical rule at $e = 0$. The last column is what the sealed records show. The two rules disagree on exactly the cells where the classical one is wrong.

cell	q_m	ρ	e	classical	ours	rules say	sealed	
Multi-IF 5 lang., turn 1	0.415	0.465	0.518	0.465	0.296	skip / pay	paid	
Multi-IF 5 lang., turn 2	0.282	0.453	0.614	0.453	0.242	skip / pay	paid	
Multi-IF 5 lang., turn 3	0.140	0.453	0.672	0.453	0.214	skip / skip	cost more	
Multi-IF En/Fr/Es, turn 2	0.226	0.451	0.315	0.451	0.360	skip / skip	cost more	
Multi-IF En/Fr/Es, turn 3	0.203	0.449	0.555	0.449	0.266	skip / skip	cost more	Classical
Multi-IF English, turn 1	0.419	0.465	0.000	0.465	0.465	skip / skip	cost more	
Multi-IF Fr/Es, turn 1	0.518	0.431	0.176	0.431	0.384	pay / pay	paid	
IFBench classic	0.491	0.452	0.126	0.452	0.419	pay / pay	paid	
IFBench out-of-dist.	0.132	0.447	0.126	0.447	0.414	skip / skip	cost more	
JSONSchemaBench GitHub	0.633	0.453	0.109	0.453	0.425	pay / pay	paid	
JSONSchemaBench other	0.793	0.407	0.000	0.407	0.407	pay / pay	paid	

index correct on 9 of 11 cells; the elasticity-corrected condition on 11 of 11. The condition was derived after these cells were sealed, so this is a fit and not a test; it is pre-registered for the cross-family ladder, whose cells were authorized before any of their comparator tokens existed. Underlining marks a wrong verdict.

replicates the portfolio produced 0 harms in 416 task-replicates, and 2 harms in 244 confirmation tasks, both truncations of a comparator proof beyond the reserved prefix rather than false certificates. Public example tests on LiveCodeBench are inexact (p_a between .72 and .98 per model in the frontier pool); with gpt-5 at $\sigma_P = .87$ the predicted harm rate $q_a(1-p_a)$ is of the same order as the rescue rate, and the audit found 4 rescues against 7 harms for the best cascade on the fit role, so that fit-role cascade fails the registered improvement gate. These finite counts do not prove impossibility for every sample size or every policy. A pre-registered pilot measures the same bound directly with the open-checkpoint certifier of the instruction-following cells: on 126 never-opened stdin problems of LiveCodeBench release v6 (fit role only), the 4B model's public-test certificate rate is $q_a = 0.45$ (exact interval $[0.36, 0.54]$), above the pre-registered 0.25–0.40 and above the cost thresholds of both comparators (0.42 at 32B, 0.36 at 72B), but its precision against the official private tests is $p_a = 0.81$ ($[0.68, 0.90]$; 11 false certificates among 57, seven of them on hard problems), so the predicted harm rate $q_a(1-p_a) = 0.087$ exceeds the one-point margin nine-fold and the pre-registered rule blocks the family before any comparator is generated. Code with public tests is therefore excluded by the harm bound, not by cost.

D CERTIFIED REFINEMENT TREES: STATEMENTS AND PROOFS

This appendix states and proves the three guarantees behind the protected refinement tree swept on the routing pools (Section 4). Notation: a released public context $\phi \in \Phi$ has the large model P; a tree T_ϕ of nested public partitions has nodes v with fit rows R_v, $n_v = |R_v|$, a root with n_0 rows, and L levels; for a candidate action $a \in \mathcal{A}$ the signed advantage on request i is $\Delta_a(i) = u_{x_i}(y_a) - u_{x_i}(y_P) \in D = \{-1, 0, +1\}$, and $\mu_v(a) = \mathbb{E}[\Delta_a(i) \mid \phi(x_i) \in v]$ is its population average on the node's public cell. The tree, the number of levels L, and the candidate sets A_v are fixed before any fit outcome is read; A_v contains the actions whose complete cap fits the request budget on every request of the node, which is a public quantity. The fit rows of a node are treated as independent draws from its public cell. Every certificate is a one-sided lower confidence bound $\ell_v(a)$ computed from the fit rows of v only.

Proposition 10 (Simultaneous validity after selection)**.** *Let the bound at node v for action $a \in A_v$ be computed at level $\delta_{v,a} = \delta \frac{n_v}{n_0} \cdot \frac{1}{L} \cdot \frac{1}{|A_v|}$, Conditional on the counts (n_v), suppose each bound satisfies $\Pr[\ell_v(a) > \mu_v(a) \mid (n_v)] \le \delta_{v,a}$. Empty nodes receive no action and zero error allocation. Then with probability at least $1-\delta$, simultaneously for every node v and every $a \in A_v$, $\ell_v(a) \le \mu_v(a)$.*

In particular the bound of the action selected by the dynamic program at any node is valid, whatever the selection rule.

Proof. Conditional on the node counts, within each level the nodes partition the root rows, so $\sum_{v\in\text{level}} \delta \frac{n_v}{n_0}\frac{1}{L} = \frac{\delta}{L}$; summing over the L levels gives δ, and summing the equal shares over A_v inside a node leaves the node total unchanged. Hence $\sum_v \sum_{a\in A_v} \delta_{v,a} = \delta$, and the union bound gives conditional simultaneous coverage. Averaging over the counts gives the unconditional assertion. Simultaneous coverage is a property of the whole collection of bounds, so it is unaffected by which bound is later chosen. □

The bound used everywhere in this paper is the exact lattice bound on the declared support $D = \{-1, 0, +1\}$. Let $\mathrm{CP}^-_\eta(k, n)$ and $\mathrm{CP}^+_\eta(k, n)$ denote the Clopper–Pearson lower and upper bounds at level η on a proportion with k successes in n trials, and let n_+ and n_- be the numbers of rescues ($\Delta = +1$) and harms ($\Delta = -1$) among the n_v rows. Then

$$\ell_v(a) \;=\; \mathrm{CP}^-_{\delta_{v,a}/3}(n_+, n_v) \;-\; \mathrm{CP}^+_{\delta_{v,a}/3}(n_-, n_v),$$

the level being split equally over the three values of D. A union bound over D gives $\Pr[\ell_v(a) > \mu_v(a)] \le \delta_{v,a}$, as required by Proposition 10. An unobserved harm is charged its full upper bound $\mathrm{CP}^+_{\delta_{v,a}/3}(0, n_v)$, which is what makes small cells expensive.

Proposition 11 (Protected-policy dominance)**.** *Let π_T be the deployed policy: at every request it returns the action of the deepest deployed node that contains the request's public context, or P when that node was pruned, is unseen, or failed the mechanism gate. For the population statement below, require the deployed cells to be pairwise disjoint, and require each bound to concern the cell on which its action is actually delivered. Write $p_v = \Pr[X \in v]$ under the fit sampling distribution. Let the dynamic program deploy v only when $\ell_v(a_v) + \varepsilon > 0$, for a fixed margin $\varepsilon \ge 0$. Then (i) on every request the realized cost of π_T is at most the request budget $B(x)$, because an action enters A_v only if its complete cap fits $B(x)$ on every request of the node and an agreement cascade calls P only under the residual budget; (ii) on the event of Proposition 10, the population advantage of π_T over P on the fit distribution is at least $\sum_{v\ \text{deployed}} p_v\, \ell_v(a_v) \ge -\varepsilon$, and at least 0 when $\varepsilon = 0$; and (iii) every context that fails the mechanism gate is replayed byte for byte as P.*

Proof. (i) is the admission rule: a call is made only if its complete cap fits the remaining budget, and the realized cost never exceeds the cap. For (ii), the deployed nodes are pairwise disjoint public cells, so the advantage decomposes as the cell-weighted sum of $\mu_v(a_v)$, each bounded below by $\ell_v(a_v)$ on the simultaneous event; the deployment rule makes every cell advantage at least $-\varepsilon$. The masses sum to at most one, and the fallback has zero advantage. The sum is nonnegative when $\varepsilon = 0$. (iii) is the fallback rule of the replay map. □

Statement (ii) concerns the fit distribution. The confirmed cells of this paper are certified on the confirmation role by their own exact bound; (ii) is not the source of any confirmed-cell guarantee. The population weights p_v are not the empirical proportions n_v/n_0. If a deepest-node policy overrides an ancestor on only part of its cell, a bound for the whole ancestor does not certify its remaining delivery region. Such a policy requires bounds for those regions; the statement does not infer them.

Proposition 12 (Fragmentation law)**.** *Fix a confidence level $1 - \delta$ and a cell of n rows showing $\lceil \gamma n \rceil$ rescues ($\Delta = +1$), for a rescue fraction $\gamma \in (0, 1]$, and no harm. With δ split equally over the declared support D, the exact lattice bound is positive only if*

$$\mathrm{CP}^-_{\delta/3}(\lceil \gamma n \rceil, n) \;>\; 1 - (\delta/3)^{1/n},$$

the right-hand side being the Clopper–Pearson upper bound on an unobserved harm probability. Consequently no certificate exists below the sample size $n^*(\gamma, \delta)$, the least n that satisfies the display; at $\delta = .05$, $n^* = 181, 91, 41, 27, 15$ for $\gamma = .05, .10, .20, .30, .50$ respectively, about three times the $\lceil \log \delta / \log(1 - \gamma) \rceil = 59, 29, 14, 9, 5$ rows needed merely to observe one rescue.

Proof. With no observed harm the only negative term of the lattice bound is $-\mathrm{CP}^+_{\delta/3}(0, n)$, and $\mathrm{CP}^+_\eta(0, n) = 1 - \eta^{1/n}$ exactly. The bound is therefore positive exactly when the display holds, and n^* is obtained by direct evaluation of the display over n. □

Remark 13 (Refinement). Refining a cell of n rows into k equal children multiplies the per-child rescue rate that a certificate needs by roughly $\sqrt{k}$ at fixed total rows, because the deficit between the observed rescue fraction and its lower bound scales like $\sqrt{k/n}$ while the unobserved-harm term scales like k/n. Proposition 12 is why the official-subtype trees of Section 4 prune to their roots: their cells hold 4 to 40 rows, below n^* for the rescue rates observed there, so the population monotonicity of refinement cannot be turned into a finite-sample certificate on those benchmarks.

E COMPLETE EXPERIMENTAL RECORD

This appendix gives the experimental record that Section 4 summarizes: every cell the main text mentions, with its protocol, controls and boundary cases. Table 6 lists every sealed cell; the generated tables use earlier names and status labels (Appendix I), and Appendix A.7 maps the vocabulary of the earlier implementation to that of the main text.

Table 6: Efficiency ledger. Accuracy difference of the sealed StateCompose policy versus the strongest registered comparator and the aggregate inference-cost ratio (parameter-weighted tokens, or USD for the API pool). Confirmed rows are closed-role results; the others are labelled.

application	n	comparator	Δ (pt)	cost ratio	status
Classification (HEADLINES, official test)	5,000	Qwen2.5-3B-Instruct external	+15.22	0.799	confirmed
Open QA (SimpleQA, frontier API pool)	1,291	gpt-5	+7.13	0.004	confirmed
Code (MBPP, LLMRouterBench Performance-20)	305	GLM-Z1-9B (protected root)	+12.79	0.055	confirmed
Theorem proving, second scale (MiniF2F-valid mechanism role)	104	Goedel-Prover-V2-32B (8,192 cap)	+4.81	0.533	mechanism
Theorem proving, second scale (MiniF2F-test, never opened)	244	Goedel-Prover-V2-32B (8,192 cap)	+4.92	0.692	development
Theorem proving, second benchmark (ProofNet-valid mechanism role)	182	Goedel-Prover-V2-32B (8,192 cap)	+11.54	0.888	mechanism gate failed
Instruction following, second scale (Multi-IF turn-1, five languages, never opened)	1,166	Qwen3-32B greedy, 1,280 cap	+6.60	0.432	confirmed
Instruction following, second scale (Multi-IF turn 2, five languages, never opened)	1,166	Qwen3-32B greedy, 1,280 cap	+8.32	0.577	confirmed
Instruction following, second scale (Multi-IF turn 1, French + Spanish, never opened)	495	Qwen3-32B greedy, 1,280 cap	+5.45	0.297	confirmed
Instruction following, second scale (Multi-IF turn 2, English + French + Spanish, never opened)	933	Qwen3-32B greedy, 1,280 cap	+6.86	0.482	confirmed
Instruction following, second scale (Multi-IF turn 3, English + French + Spanish, never opened)	912	Qwen3-32B greedy, 1,280 cap	+4.50	0.654	confirmed
Instruction following, second scale (Multi-IF turn 3, mechanism role)	723	Qwen3-32B greedy, 1,280 cap	+4.29	0.676	mechanism gate failed
Instruction following, second scale (IFBench multi-turn, classic constraints, never opened)	861	Qwen3-32B greedy, 2,048 cap	+4.88	0.308	confirmed
Instruction following, second scale (IFBench multi-turn, OOD constraints, mechanism role)	409	Qwen3-32B greedy, 2,048 cap	+5.13	0.712	mechanism gate failed
Structured output, second scale (JSONSchemaBench GitHub easy+medium+hard, never opened)	731	Qwen3-32B greedy, 2,048 cap	+1.64	0.355	confirmed
Structured output, second scale (JSONSchemaBench other collections, never opened)	534	Qwen3-32B greedy, 2,048 cap	+0.75	0.274	confirmed

application	n	comparator	Δ (pt)	cost ratio	status
Instruction following, second comparator (Multi-IF five languages, turn 1, P2; opened subset, fresh 72B arm)	1,166	Qwen2.5-72B-Instruct greedy, 1,280 cap	$+5.83$	0.362	confirmed
Instruction following, second comparator (Multi-IF French + Spanish, turn 1, P2; opened subset, fresh 72B arm)	495	Qwen2.5-72B-Instruct greedy, 1,280 cap	$+4.44$	0.212	confirmed
Instruction following, second comparator (IFBench classic constraints, P2; opened subset, fresh 72B arm)	861	Qwen2.5-72B-Instruct greedy, 2,048 cap	$+3.60$	0.240	confirmed
Instruction following, second comparator (IFBench OOD constraints, P2; opened subset, fresh 72B arm)	692	Qwen2.5-72B-Instruct greedy, 2,048 cap	$+9.25$	0.633	confirmed
Structured output, second comparator (JSONSchemaBench GitHub, P2; opened subset, fresh 72B arm)	731	Qwen2.5-72B-Instruct greedy, 2,048 cap	$+1.78$	0.289	confirmed
Structured output, second comparator (JSONSchemaBench other, P2; opened subset, fresh 72B arm)	534	Qwen2.5-72B-Instruct greedy, 2,048 cap	$+1.50$	0.209	confirmed
Instruction following, second comparator (Multi-IF five languages, turn 2, P2; opened subset, fresh 72B arm)	1,166	Qwen2.5-72B-Instruct greedy, 1,280 cap	$+10.29$	0.512	confirmed
Instruction following, second comparator (Multi-IF English + French + Spanish, turn 2, P2; opened subset, fresh 72B arm)	933	Qwen2.5-72B-Instruct greedy, 1,280 cap	$+8.90$	0.384	confirmed
Instruction following, second comparator (Multi-IF English + French + Spanish, turn 3, P2; opened subset, fresh 72B arm)	912	Qwen2.5-72B-Instruct greedy, 1,280 cap	$+8.99$	0.616	confirmed
Instruction following, second comparator (Multi-IF five languages, turn 3, P2; opened subset, fresh 72B arm)	1,159	Qwen2.5-72B-Instruct greedy, 1,280 cap	$+9.75$	0.715	failed-closed
Instruction following, second comparator (IFBench OOD constraints, P3, mechanism role)	409	Qwen2.5-72B-Instruct greedy, 2,048 cap	$+9.29$	0.822	mechanism gate failed
Theorem proving (MiniF2F-valid confirmation role, 128 tasks)	128	DeepSeek-7B code-first; Goedel root budget	$+4.69$	0.443	confirmed
Instruction following (IFEval-es, independent confirmation)	412	Qwen3-4B external	$+8.25$	0.136	confirmed
Instruction following (IFEval-en, frozen transfer replay)	424	Qwen3-4B external	$+4.95$	0.839	opened transfer replay
Instruction following (M-IFEval fr/ja, prospective confirmation)	407	Qwen3-4B external	$+7.62$	0.796	confirmed
NLI (SNLI official test)	9,824	Qwen3-8B chat (registered strict per-item control)	$+2.76$	0.904	confirmed under its own composition-superiority gate
Planning (RCPSP pool B, 10k paired streams)	10,000	Large (3B) comparator	$+1.48$	0.845	validation
Planning (RCPSP pool A, 10k paired streams)	10,000	Large (3B) comparator	$+3.34$	0.767	independent role

This appendix gives, for every cell that Section 4 cites, the number of requests in each role, the rescue and harm counts, the exact lower bound ℓ of Definition 2, and the cost ratio Γ_n of Definition 1,

Table 5: Optimization problems as each paper states them; locators in Appendix F. "none": no cost constraint is stated.

method	objective	cost constraint	stop or routing signal
FrugalGPT	expected reward	expected cost $\leq b$	learned answer score, threshold
Cascade-routing	expected estimated quality	expected cost $\leq B$	quality minus λ·cost, per step
RouteLLM, Hybrid, Avengers-Pro	fitted router, knob on cost	none	learned request score, threshold
AutoMix	performance minus λ·cost	none	self-verification score, threshold
C3PO	agreement with the strongest model	$\Pr[\text{request cost} > C^*] \leq \alpha$	own confidence score, threshold
ThriftLLM	ensemble correctness probability	subset cost $\leq B$ per request	aggregation with adaptive pruning
Certificate stopping (ours)	delivered accuracy, (1)	cost $\leq C_P(x)$ on every request	an exact verifier accepts

the policy's average cost divided by the large model's average cost on the same requests, followed in parentheses by the cost ratio on each of the two fixed halves of the role. Unless a paragraph says otherwise, the large model is Qwen3-32B, the certifier is Qwen3-4B-Instruct-2507, decoding is greedy, cost is parameter-weighted tokens, and the verifier is the benchmark's official checker. Every confirmation role was opened once; a development number comes from a mechanism or validation role. The status vocabulary and the earlier names used by the generated tables are those of Appendix I. The best-of-k sampling control, the uncapped control, the cross-family ladder, the harm bound and the hybrid policy are tabulated in Appendix F; the certifier ablation is described in Appendix I.

E.1 Exact-verifier cells at the 32B large model

Multi-IF turn 1 in five languages. The policy is confirmed on the turn-1 prompts of Multi-IF in Chinese, Hindi, Italian, Portuguese and Russian. The cell holds 2,345 prompts in 548 prompt groups, after excluding every group that an earlier Multi-IF cell had opened and every prompt of the opened M-IFEval list, split by group into the fit role (453 prompts), the mechanism role (726 prompts, two halves) and the confirmation role (1,166 prompts). The policy has no fitted parameter: the certifier answers greedily under a 1,280-token generation cap, the official Multi-IF checker is the verifier, and after a rejection the large model's own greedy answer is delivered truncated at the largest prefix that fits the remaining budget, the budget being the large model's complete cost cap. On the mechanism role the policy certifies $626/726$ against $547/726$ for the large model ($+10.9$ points; 83 rescues, 4 harms; $\ell = +7.5$ points) at cost ratio 0.427 ($0.382/0.474$) with zero violations, and the gate passes. On the confirmation role the policy attains $1{,}001/1{,}166 = 85.85\%$ against $924/1{,}166 = 79.25\%$ ($+6.60$ points; 80 rescues, 3 harms; $\ell = +4.57$ points; paired bootstrap $[+5.1, +8.1]$; cluster bootstrap $[+4.9, +8.5]$; exact $p = 2 \cdot 10^{-20}$) at cost ratio 0.432 ($0.434/0.429$; cluster bootstrap $[0.39, 0.47]$) with zero violations; the certifier's acceptance rate is 76.8%, and the large model ran on 270 prompts, on 72 of which its answer was cut off by the remaining budget. Both gates pass, and every language is positive, with cost ratio from 0.30 (Chinese) to 0.57 (Hindi). The fixed alternatives under the same budget are the two models themselves: the certifier alone certifies $896/1{,}166 = 76.8\%$ at cost ratio 0.13 and fails the non-inferiority margin by 2.4 points, the large model alone is the reference, and calling both on every request costs 1.13 times the large model. A router trained on these prompts would have to learn the checker's verdict, which the policy reads directly. The cost identity of Theorem 1 reads $0.13 + 0.23 \cdot 1.37 \cdot 0.94 = 0.43$. The prompts are translations of English prompts that earlier 4B-scale cells had opened in English; no output of either action existed on them. Under three seeds the policy attains 85.42% against 79.07%, with $\ell = +5.90$ points and cost ratio 0.4296 with zero violations.

Multi-IF turn 2 in five languages. The second turn of the same conversations is its own cell: the request is the official second user message on top of the first message and the large model's own sealed greedy first answer, identical for the policy and for the large model, with the official cumulative constraints; the input cap is 4,096 tokens and everything else is unchanged. On the mechanism role the policy certifies $496/726$ against $432/726$ ($+8.8$ points; 69 rescues, 5 harms; $\ell = +5.6$) at cost ratio 0.545 ($0.523/0.568$) with zero violations. On the confirmation role it attains $780/1{,}166 = 66.90\%$ against $683/1{,}166 = 58.58\%$ ($+8.32$ points; 102 rescues, 5 harms; $\ell = +6.01$; cluster bootstrap $[+6.4, +10.3]$; $p = 10^{-24}$) at cost ratio 0.577 ($0.585/0.569$) with zero violations and a certifier acceptance rate of 58.7%; both gates pass. With cumulative constraints

the certifier and the large model pass at almost the same rate (58.7% against 58.6%), so the saving comes from the certificate, not from a weaker task. The third turn of this series fails the cost ceiling and is described with the other ceiling failures below.

Multi-IF turn 1 in French and Spanish. The French and Spanish turn-1 prompts are translations different from those of the opened M-IFEval list, so they form a second never-opened turn-1 cell at the same disclosure level: 990 rows in the same group-wise roles (fit 186, mechanism 309, confirmation 495), with the policy, caps and gates inherited unchanged. The English turn-1 prompts are the opened IFEval originals and were generated for the large model's history only. On the mechanism role the policy certifies $282/309$ against $264/309$ ($+5.8$ points; 18 rescues, no harm; $\ell = +2.0$) at cost ratio 0.314 ($0.276/0.356$) with zero violations and a certifier acceptance rate of 81.9%, and the gate passes. On the confirmation role it attains $458/495 = 92.53\%$ against $431/495 = 87.07\%$ ($+5.45$ points; 28 rescues, 1 harm; $\ell = +2.44$ points; cluster bootstrap $[+3.0, +8.2]$; exact $p = 10^{-7}$) at cost ratio 0.297 ($0.272/0.325$; cluster bootstrap $[0.25, 0.35]$) with zero violations; the certifier's acceptance rate is 83.8%, and the large model ran on 80 prompts, on 7 of which its answer was cut off. Both gates pass, with cost ratio 0.30 on French and 0.29 on Spanish.

Multi-IF turns 2 and 3 in English, French and Spanish. The second turn of these conversations, with English added (its turn-1 prompts are the opened IFEval originals and served as history only), is its own cell of 1,856 rows. On the mechanism role the policy certifies $447/567$ against $406/567$ ($+7.2$ points; 45 rescues, 4 harms; $\ell = +3.8$) at cost ratio 0.422 ($0.393/0.453$) with a certifier acceptance rate of 72.7%. On the confirmation role it attains $703/933 = 75.35\%$ against $639/933 = 68.49\%$ ($+6.86$ points; 67 rescues, 3 harms; $\ell = +4.49$; cluster bootstrap $[+5.1, +8.7]$; $p = 10^{-16}$) at cost ratio 0.482 ($0.494/0.468$; cluster bootstrap $[0.43, 0.53]$) with zero violations; the certifier's acceptance rate is 67.7%, and the large model ran on 301 rows, on 43 of which its answer was cut off. Both gates pass, with cost ratio 0.47 on English, 0.49 on French and 0.50 on Spanish. The third turn (1,815 rows; 41 conversations without a third turn are dropped) passes its gate where the five-language series stopped. On the mechanism role the policy certifies $377/554$ against $349/554$ ($+5.1$ points; 34 rescues, 6 harms; $\ell = +1.7$) at cost ratio 0.553 ($0.507/0.602$) with a certifier acceptance rate of 62.1%. On the confirmation role it attains $548/912 = 60.09\%$ against $507/912 = 55.59\%$ ($+4.50$ points; 47 rescues, 6 harms; $\ell = +2.21$; cluster bootstrap $[+2.9, +6.2]$; $p = 6 \cdot 10^{-9}$) at cost ratio 0.654 ($0.671/0.635$; cluster bootstrap $[0.60, 0.70]$) with zero violations; the certifier's acceptance rate is 51.2%, and the large model ran on 445 rows, on 78 of which its answer was cut off. Both gates pass, the improvement gate at a relative error reduction of 10.1%, just above its bar; the cost ratio is 0.63 on English, 0.65 on French and 0.71 on Spanish, and Spanish is a language, not a half, so it is not a gate condition. Across this group the acceptance rate falls from 0.84 to 0.68 to 0.51 and the cost ratio rises from 0.30 to 0.48 to 0.65, the same law as in the five-language series, with the third turn landing just under the ceiling instead of just over it.

IFBench classic constraints. IFBench's multi-turn release pairs a held-out WildChat prompt and the released assistant answer with a second user message, "Rewrite your response" plus one verifiable constraint; the released exchange is the public history, identical for the policy and for the large model, and the official prompt-level loose check is the verifier. Two subsets are separate cells with the same policy, a 2,048-token generation cap and roles assigned per WildChat prompt (732 prompts appear in both subsets and keep one role): the classic-constraint subset (1,774 rows) and the out-of-distribution subset (1,387 rows, 33 new constraint types). The single-turn IFBench test set had been opened by earlier small-scale cells and is not claimed. Scoring the classic subset exposed a defect of the release: 73 rows carry checker arguments that the official checker cannot evaluate deterministically (72 letter-frequency rows omit the comparison relation, which the checker then draws at random, and one row lacks its paragraph index and first word). Before any confirmation outcome existed, those rows were excluded from both the policy and the large model by a rule computed from the request list and the checker source alone; the retained rows reproduce the earlier scoring except three row-actions of the two case-change checkers, which call an unseeded language detector. On the mechanism role the policy certifies $467/523$ against $443/523$ ($+4.6$ points; 24 rescues, no harm; $\ell = +2.1$) at cost ratio 0.354 ($0.379/0.329$) with zero violations and a certifier acceptance rate of 79.0%, and the gate passes. On the confirmation role it attains $757/861 = 87.92\%$ against $715/861 = 83.04\%$ ($+4.88$ points; 42 rescues, no harm; $\ell = +2.96$ points; cluster bootstrap $[+3.5, +6.4]$; exact $p = 5 \cdot 10^{-13}$) at cost ratio 0.308 ($0.289/0.327$; cluster bootstrap $[0.28, 0.34]$)

with zero violations; the certifier's loose acceptance rate is 80.7%, and the large model ran on 166 rows, on 3 of which its answer was cut off. Both gates pass, and the strict metric moves the same way (85.7% against 81.4%). Under three seeds the policy attains 87.69% against 83.62%, with $\ell = +3.84$ points and cost ratio 0.3047 with zero violations. The out-of-distribution subset fails the cost ceiling and is described with the other ceiling failures below.

JSONSchemaBench GitHub. JSONSchemaBench asks for a JSON object that conforms to a real-world schema; the official metric is prompt-level schema compliance under Draft 2020-12 with the format checker, and that function, applied to the certifier's completion, is the verifier. The cell pools the three GitHub test splits (easy, medium, hard). The prompt reproduces the official task for chat models: the task description as the system message, its two few-shot samples as prior turns, and the official blank-line stop rule applied before scoring. The requests are fixed by four schema-only rules sealed before any outcome: 106 hard schemas longer than 16,000 characters, 14 schemas using the formats ipv4, ipv6 or uuid (the official checker's functions for those formats reject every instance) and 3 duplicates are excluded, leaving 1,411 rows (fit 265, mechanism 415, confirmation 731). The generation cap is 2,048 new tokens under an 8,192-token input cap that no row reaches (the rendered prompts are at most 4,785 tokens under both tokenizers). On the mechanism role the policy certifies $386/415$ against $373/415$ ($+3.1$ points; 13 rescues, no harm; $\ell = +0.6$) at cost ratio 0.394 (0.415/0.375) with a certifier compliance rate of 81.0%, and the gate passes. On the confirmation role it attains $691/731 = 94.53\%$ against $679/731 = 92.89\%$ ($+1.64$ points; 12 rescues, no harm; $\ell = +0.24$ points; cluster bootstrap $[+0.8, +2.6]$; exact $p = 5 \cdot 10^{-4}$) at cost ratio 0.355 (0.363/0.348; cluster bootstrap $[0.32, 0.39]$) with zero violations; the certifier's compliance rate is 83.2%, the large model ran on 123 rows, on 3 of which its answer was cut off, and no validation timed out. Both gates pass, with cost ratio 0.20 on easy, 0.35 on medium and 0.50 on hard. Under three seeds the policy attains 93.98% against 92.61%, with $\ell = +1.11$ points and cost ratio 0.3656 with zero violations. The cost identity reads $\alpha = 0.12$, $q_1 = 0.83$ and $\kappa \cdot \tau = 1.37$: the large model writes long objects exactly on the schemas that the certifier fails.

JSONSchemaBench other collections. The benchmark's seven other released collections form a second cell under the same prompt, checker and rules, with 1,067 of 1,324 rows after the outcome-free exclusions: the 16,000-character rule removes all fifty *ultra* schemas and 166 others, mostly Kubernetes and JsonSchemaStore; 37 Kubernetes schemas are exact duplicates; seven JsonSchemaStore identifiers repeat; and four schemas use a format that the checker rejects. On the mechanism role the policy certifies $303/309$ against $300/309$ ($+1.0$ points; 3 rescues, no harm) at cost ratio 0.246 (0.244/0.248) with a certifier compliance rate of 90.6%. On the confirmation role it attains $521/534 = 97.57\%$ against $517/534 = 96.82\%$ ($+0.75$ points; 4 rescues, no harm; $\ell = -0.58$ points, inside the one-point margin; cluster bootstrap $[+0.2, +1.5]$) at cost ratio 0.274 (0.238/0.305; cluster bootstrap $[0.23, 0.32]$) with zero violations; the certifier's compliance rate is 89.0%, and the large model ran on 59 rows, on 1 of which its answer was cut off. The efficiency gate passes and the improvement gate does not (exact $p = 0.125$), the expected shape when the large model is already close to perfect compliance. The improvement gate fails for want of disagreement rather than want of effect: the policy and the large model differ on four requests and the policy wins all four; an exact two-sided test on a split of 4 rescues against 0 harms returns $p = 0.125$, and no split with fewer than six discordant requests can reject at the 5% level however lopsided it is (5 against 0 gives 0.0625, 6 against 0 gives 0.03125). At the realized rescue rate of $4/534$ a cell would need roughly 800 requests to expect six such disagreements and about $1{,}300$ to expect ten, and the collection is exhausted at $1{,}067$ usable rows. The cell is reported as failing the gate it fails; re-running it would return the same answer, since generation is greedy and the role has been opened. Per collection the cost ratio runs from 0.18 (Glaiveai2K) to 0.49 (JsonSchemaStore).

Sampling control. The best-of-k sampling control of Section 4 is reported in Table 14, Appendix F.

Cells sealed before the efficiency gate. Six cells were sealed before the efficiency gate of Definition 2 existed and carry their own earlier gates (Appendix I); the 128-task MiniF2F cell and the MBPP root are described below. The earlier gate requires at least a $+2$-point gain over the large model, a 5% relative error reduction, positive effects on both halves, a positive paired lower bound, a Holm-significant exact discordance test, and per-request and average cost feasibility; its

inference uses 10,000 paired bootstraps, deterministic halves, paired exact discordance tests and Holm correction over the registered comparison family. On Spanish IFEval (IFEval-es) the policy attains $359/412 = 87.14\%$, against $325/412 = 78.88\%$ for Qwen3-4B, the large model, and for the strongest same-family policy; its certifiers, in order, are Qwen3-1.7B, two samples of LFM2.5-1.2B and one sample of Qwen3-0.6B. It beats the large model by $+9.19$ and $+7.49$ points on the two halves and reduces error by 39.08%; against the strongest same-family policy it yields 38 rescues and four harms (Holm-adjusted exact $p = 1.13 \cdot 10^{-7}$). Its maximum per-request cost ratio is .9999 and its average cost ratio is .1360; the same-family policy's are .9965 and .1495, so both respect the cap and only the policy is cheaper on average (Table 7). On M-IFEval in French and Japanese the frozen policy attains $320/407 = 78.62\%$, against 71.01% for Qwen3-4B and 71.50% for the same-family control assigned in advance ($+7.62$ points over Qwen3-4B; clustered interval $[11.46, 39.39]\%$ on the relative error reduction); it is positive on both halves of both languages, removing the certifier from the other family lowers it to 65.36%, the 53/22 rescue/harm contrast against Qwen3-4B and the 54/0 contrast against that removal survive the frozen inference family, and the policy saves 20.36% of Qwen3-4B's parameter-weighted tokens. On the HEADLINES test set the frozen policy attains 79.42%, against 72.62% for the best single open model ($+6.80$ points; interval $[5.64, 7.96]$) and 64.20% for the registered external 3B large model; removing the certifier from the other family lowers it to 77.64% (a loss of 1.78 points; interval $[0.94, 2.64]$) with positive effects on both halves, and the policy saves 20.06% against the external 3B large model. On the 9,824 consensus-quality rows of the SNLI test set the policy raises accuracy from 77.85% for Qwen3-8B to 80.61% ($+2.76$ points; independent clustered lower bound $+2.24$ points); a control that composes Qwen3-8B with itself under the same public state reaches 79.99% but costs 9.57 to 9.70% more. The SNLI cell passed a composition-superiority gate, not the efficiency gate, and is not counted (Section 3). Table 7 lists the IFEval-es cell, the RCPSP validation cells and two diagnostics with their paired intervals.

Proof-checker cells. The formal-proof policy calls DeepSeek-Prover-V2-7B under a code-first 768-token protocol, stops on an isolated Lean 4.16 certificate, and otherwise admits Goedel-Prover-V2-8B only when its complete cost cap fits. On the 128-task MiniF2F cell it proves $91/128$ tasks, against $85/128$ for DeepSeek-Prover-V2-7B code-first alone and $74/128$ for Goedel-Prover-V2-8B code-first alone; all six gains over the strongest fixed protocol are Goedel-Prover-V2-8B rescues and none is a loss (gains of $+2$ and $+4$ on the two halves; bootstrap interval $[1.56, 8.59]$ points; Holm $p = .03125$), and the maximum and average cost ratios against the large model are .891 and .443. A second cell was then run on MiniF2F-test, a role never opened before, with Goedel-Prover-V2-32B under an 8,192-token cap as the large model: the policy runs DeepSeek-Prover-V2-7B code-first, stops on the isolated certificate, and otherwise replays the large model's own sample under the remaining budget. It certifies $154/244$ against $142/244$ for the large model alone (14 rescues, 2 harms; halves $80/75$ and $74/67$): $+4.92$ points, 11.8% relative error reduction, $\ell = +0.24$ points, paired bootstrap $[+2.05, +8.20]$ points, exact $p = 0.0042$, at cost ratio 0.692 (bootstrap $[0.631, 0.751]$; halves 0.679/0.704) with zero violations. Neither gate is awarded. The bound $\ell = +0.24$ points holds under a two-way split of δ over the observed values and changes sign under the three-way split over the declared support used everywhere else, so the improvement gate is not claimed; the efficiency gate's condition that both halves stay at or below 0.70 is missed on one half at 0.704, and it is reported as failed rather than relaxing the ceiling. The cell is a development row of Table 6. Its mechanism role had given $80/104$ against $75/104$ at cost ratio 0.533. Across the sealed mechanism seed and three fresh registered seeds on the same 104 tasks, the cost ratio is 0.517 ± 0.044 and the net gain over the large model is $+8.0 \pm 2.0$ tasks with 0 harms; the seed standard deviation 0.044 exceeds the 0.008 by which the MiniF2F-test cost ratio 0.692 lies below the 0.70 ceiling. ProofNet, the cell registered next under the same gate, fails on acceptance rate and is described with the failure modes below.

E.2 The 72B large model

Second large model: Qwen2.5-72B-Instruct. Qwen2.5-72B-Instruct (72.7B parameters, bf16, greedy decoding with repetition penalty 1.0, and its chat template's default system prompt) replaces Qwen3-32B on the opened instruction-following and structured-output cells. Roles are inherited from the 32B series; on the turn-1 cells the certifier's and the 32B's outputs and outcomes are the sealed ones and are reused unchanged, only the 72B answers being generated; on the later turns every answer is generated afresh, because the requests carry the 72B's own earlier answers; and each

Table 7: Sealed results. Each accuracy cell is CARP/Large/Same; differences are percentage points with paired 95% intervals. I, V, and D denote independent role, prospective validation, and diagnostic; "Gate–" denotes a failed preregistered headline gate, and "mech+" only a passed residual-mechanism gate. Cost is maximum-request/aggregate model-inference ratio to the Large cap, with admitted/realized semantics labeled when they differ.

Domain (role)	Acc. C/L/S	ΔLarge [CI]	ΔSame [CI]	Ledger max/total
IFEval-es 41-policy finite $n = 412$, I-pass	C 87.14 L 78.88 S 78.88	$+8.25$ $[4.37, 12.14]$	$+8.25$ $[5.34, 11.41]$	PWT .9999/.1360 FLOP .9782/.1319
IFBench fixed 4-edge $n = 167$, D-post	C 55.69 L 31.14 S 55.69	$+24.55$ $[16.77, 32.34]$	0.00 $[-2.40, 2.40]$	FLOP-real .965/.338
RCPSP A paired streams $n = 10$k, I-Gate–	C 7.93 L 4.59 S 6.06	$+3.34$ $[2.79, 3.90]$	$+1.87$ $[1.47, 2.28]$	PWT .841/.767
RCPSP B paired streams $n = 10$k V-mech+	C 6.00 L 4.52 S 4.47	$+1.48$ $[1.02, 1.95]$	$+1.53$ $[1.29, 1.77]$	PWT .974/.845
IFEval midscale 41-policy finite $n = 412$, D	C 85.92 L 78.64 S 77.18	$+7.28$ $[3.64, 11.17]$	$+8.74$ $[5.83, 11.65]$	FLOP .985/.212

cell's cost ratio was registered in advance from the corresponding 32B cell through Proposition 2 ($\alpha_{72} = \alpha_{32} \cdot N_{32}/N_{72}$ with the sealed q_1 and $\kappa \cdot \tau$). Two policies are sealed together: P2, the certifier and then the 72B under the 72B's budget, and P3, the certifier, then the sealed 32B run in full as a middle stage, then the 72B under the remaining budget. The confirmation role of a cell opens at most once, when P2's mechanism gate passes; P3's confirmation is computed only where its own gate passed. Table 8 gives the confirmed cells next to their 32B counterparts. On the five cells whose 32B confirmation had been sealed (Multi-IF turn 1 in five languages and in French and Spanish, IFBench classic constraints, both JSONSchemaBench cells) every P2 mechanism gate passes at cost ratio 0.19 to 0.34 (the English turn-1 rows, opened only to supply history, pass too and carry no claim), and every P2 confirmation passes the efficiency gate at cost ratio 0.21 to 0.36 with zero violations, the improvement gate passing in four of five (JSONSchemaBench's other collections, with the 72B at 94.2%, misses it by 0.16 points of ℓ). Across those five cells the realized P2 cost ratio differs from the registered one by at most 0.011 (French and Spanish, 0.212 against 0.223); Multi-IF turn 1 in five languages, IFBench classic constraints and both JSONSchemaBench cells agree to 0.003 or better. The later-turn cells were then run afresh: P2 is confirmed on five-language turn 2 and on English, French and Spanish turns 2 and 3 at $+10.29$, $+8.90$ and $+8.99$ points and cost ratios 0.512, 0.384 and 0.616, while five-language turn 3 passes its P2 mechanism gate at cost ratio 0.628 and fails its confirmation at 0.715 despite $+9.75$ points, the largest mechanism-to-confirmation cost gap in the series, and is frozen (Table 8). P3 is confirmed on the same five turn-1 cells at cost ratio 0.16 to 0.34 with both gates, its realized ratios at or below the registered values (the 32B stage rescues 0.37 to 0.78 of the certifier's rejections and costs less than the 72B call it replaces). The registration named three cells where P3 would fail the 0.70 ceiling, and it is right on two of them. On IFBench's out-of-distribution subset the P2 mechanism gate passes at cost ratio 0.651 ($0.667/0.632$; registered 0.64) while P3 reaches 0.822 (registered 0.83) and is frozen, and five-language turn 3 is frozen under P3 on the per-half condition at $0.630/0.710$ although its overall cost ratio 0.669 is under the ceiling. On English, French and Spanish turn 3 the prediction is falsified: that cell is confirmed with both gates at cost ratio 0.655 against a registered 0.764. The P2 confirmation of IFBench's out-of-distribution subset, opened once, passes both gates on its 692 never-opened rows: $411/692 = 59.4\%$ against $347/692 = 50.1\%$ for the 72B ($+9.2$ points; 64 rescues, no harm; $\ell = +6.4$; exact $p = 10^{-19}$) at cost ratio 0.633 ($0.614/0.651$; cluster bootstrap $[0.59, 0.68]$) with a certifier acceptance rate of 46%: the cell that the 32B large model failed at cost ratio 0.71 is confirmed at the 72B because α falls from 0.13 to 0.06 while q_1 and $\kappa \cdot \tau$ are unchanged. The certifier's answers on those confirmation rows did not exist in the 32B series, where the cell was frozen at the mechanism stage, so they were generated under an amendment sealed before the replay that adds only those answers and changes no cap, seed or gate.

Universe	n	Qwen3-32B comparator				Qwen2.5-72B comparator, P2 (pred.)				
		q_1	ratio (halves)	Δ (LCB)	verdict	q_1	ratio (halves)	pred.	Δ (LCB)	verdict
Multi-IF turn 1, five languages	1,166	77%	0.43 (0.43/0.43)	+6.6 (+4.6)	confirmed	77%	0.36 (0.37/0.36)	0.36	+5.8 (+3.7)	confirmed
Multi-IF turn 2, five languages	1,166	59%	0.58 (0.58/0.57)	+8.3 (+6.0)	confirmed	58%	0.51 (0.52/0.50)	0.51	+10.3 (+7.9)	confirmed
Multi-IF turn 3, five languages (32B: mech.)	1,159	50%	0.68 (0.64/0.71)	+4.3 (+1.5)	gate failed	39%	0.71 (0.74/0.69)	–	+9.7 (+7.4)	failed
Multi-IF turn 1, French and Spanish	495	84%	0.30 (0.27/0.32)	+5.5 (+2.4)	confirmed	84%	0.21 (0.20/0.23)	0.22	+4.4 (+1.6)	confirmed
Multi-IF turn 2, Eng/Fr/Sp	933	68%	0.48 (0.49/0.47)	+6.9 (+4.5)	confirmed	68%	0.38 (0.41/0.36)	0.41	+8.9 (+6.0)	confirmed
Multi-IF turn 3, Eng/Fr/Sp	912	51%	0.65 (0.67/0.63)	+4.5 (+2.2)	confirmed	48%	0.62 (0.63/0.60)	0.59	+9.0 (+6.4)	confirmed
IFBench, classic constraints	861	81%	0.31 (0.29/0.33)	+4.9 (+3.0)	confirmed	81%	0.24 (0.22/0.26)	0.24	+3.6 (+1.8)	confirmed
IFBench, OOD (32B: mech.)	692	43%	0.71 (0.73/0.69)	+5.1 (+2.1)	gate failed	46%	0.63 (0.61/0.65)	–	+9.2 (+6.4)	confirmed
JSONSchemaBench, GitHub	731	83%	0.36 (0.36/0.35)	+1.6 (+0.2)	confirmed	83%	0.29 (0.30/0.28)	0.29	+1.8 (+0.3)	confirmed
JSONSchemaBench, other	534	89%	0.27 (0.24/0.30)	+0.7 (−0.6)	confirmed	89%	0.21 (0.17/0.24)	0.21	+1.5 (−0.2)	confirmed

P3 (4B, then the sealed 32B in full, then the 72B prefix); role-specific rows

Universe	n	q_1	ratio (halves)	pred.	Δ (LCB)	verdict
Multi-IF turn 1, five languages	1,166	77%	0.34 (0.33/0.34)	0.40	+7.9 (+5.5)	confirmed
Multi-IF turn 2, five languages	1,166	58%	0.53 (0.54/0.52)	0.67	+14.8 (+12.1)	confirmed
Multi-IF turn 3, five languages (32B: mech.)	723	50%	0.67 (0.63/0.71)	0.81	+15.1 (+11.4)	gate failed
Multi-IF turn 1, French and Spanish	495	84%	0.19 (0.17/0.21)	0.22	+7.3 (+3.9)	confirmed
Multi-IF turn 2, Eng/Fr/Sp	933	68%	0.40 (0.42/0.38)	0.51	+12.4 (+9.4)	confirmed
Multi-IF turn 3, Eng/Fr/Sp	912	48%	0.66 (0.65/0.66)	0.76	+14.4 (+11.1)	confirmed
IFBench, classic constraints	861	81%	0.25 (0.23/0.27)	0.26	+6.2 (+3.9)	confirmed
IFBench, OOD (32B: mech.)	409	43%	0.82 (0.86/0.78)	0.83	+9.3 (+4.8)	gate failed
JSONSchemaBench, GitHub	731	83%	0.23 (0.24/0.22)	0.23	+3.7 (+1.6)	confirmed
JSONSchemaBench, other	534	89%	0.16 (0.12/0.20)	0.17	+3.7 (+1.4)	confirmed

Table 8: Second large model. Sealed 32B and 72B results, with role-specific denominators specified in the accompanying paragraph; q_1 is the certifier's acceptance rate, ratios are average parameter-weighted tokens divided by the large model's (halves in parentheses), "pred." is the cost ratio registered in advance from the 32B cell, and Δ is the accuracy difference in points with its exact lower bound. For the out-of-distribution IFBench subset the 32B series never opened the confirmation role, so its 32B columns are the frozen mechanism role (409 rows), its 72B prediction (0.64) was registered from that role and is shown as absent, and its P3 row is the frozen 72B mechanism result.

Role-specific denominators for Table 8. The displayed n in the P2 block belongs to the 72B P2 result, not necessarily to the adjacent 32B result. For Multi-IF turn 3 in five languages, the 32B result uses 723 mechanism requests, 72B P2 uses 1,159 confirmation requests, and 72B P3 uses 723 mechanism requests. For the out-of-distribution IFBench subset, these roles and counts are, respectively, mechanism 409, confirmation 692 and mechanism 409. Comparisons across these roles are not paired comparisons on a common sample. The generated table is retained unchanged; these denominators qualify its adjacent columns.

E.3 Routing pools without a verifier

MBPP on Performance-20. Without a verifier the root rule of Section 2 fixes one action per released context. A policy fitted on one role, sign- and cost-gated on a second, and confirmed on 2,406 independent tasks changes the action on one context only: on MBPP it replaces GLM-Z1-9B, the large model of that pool, by Qwen2.5-Coder-7B. On the 305 MBPP tasks it produces 62 rescues and 23 harms, raising accuracy from 64.92% to 77.72% ($+12.79$ points; halves $+12.18$/$+13.42$ points) at cost ratio $.0553$; across all ten contexts it reaches 69.58% against 67.96% ($+1.62$ points; interval $[.87, 2.37]$ points), with maximum per-request cost ratio one. On the validation role the official trained AvengersPro router is 2.23 points more accurate per request than this policy before the budget is applied. In this comparison the budget of each request is the realized cost of GLM-Z1-9B on that request; AvengersPro exceeds that budget on 861 responses when its 7B embedding call is charged zero, as Table 9 charges every router forward pass, and on 1,097 responses when that call is charged. Its response-safe wrapper, fitted on the fit role only, is exactly the same action sequence as ours. Under the per-request cap of Table 9 AvengersPro exceeds no budget; there it reaches 68.27% macro accuracy at 1.203 times the average cost of the large model, and charging the embedding call raises that ratio to 1.228 (Table 10). The result establishes a cheaper fixed action on one context, not a universal advantage of routing by task identity.

The refinement-tree sweep on Performance-20. A registered refinement tree over the official released subtypes of Performance-20, with single-model and agreement-cascade actions, exact simultaneous lower bounds under size-weighted alpha spending, and a bottom-up stop-versus-split rule (Appendix D), certifies no node in either gate mode (1,629 and 973 admissible hypotheses) and returns the large model alone. Under the realized per-request rule only 0 to 5 of the 19 alternatives

are admissible per context, because the large model's completion length varies by two orders of magnitude within one task type; under the per-request cap plus average budget used elsewhere in this paper 10 to 16 alternatives are admissible, yet the fit-selected task-level switch gains at most $+4.4$ points on the mechanism role. The identifiable per-request state is candidate agreement: when two proposals agree, the delivered answer is correct 84 to 97% of the time and the agreement cascade gains $+3$ to $+6$ points, but every accurate cheap proposal in this pool is itself a thinking model, so the cascade costs 1.9 to 5.0 times the large model. An exact certificate with zero observed harms needs $n^*(\gamma, \delta)$ of 15 to 181 rows per cell at $\delta = .05$ for rescue rates of 50 to 5% (Proposition 12), while official subtype cells hold 4 to 40 rows. Breadth on this pool is therefore limited by identifiability and by the pool's cost structure, not by the number of models.

SimpleQA on Performance-Cost. The same procedure was then run, unchanged, on LLMRouterBench's official Performance-Cost pool: 13 API models with the benchmark's published USD prices, where price ratios between models exceed 40 while accuracy gaps are a few points. The fit role selects gpt-5 as the large model (macro accuracy 69.88%). Under the per-request cap plus average budget the registered root rule with complete cascade actions passes the disjoint mechanism gate on four of nine fit-changed contexts. The validation role, opened once for six models and four datasets, keeps exactly one cell: on SimpleQA the non-thinking qwen3-235b-a22b-2507 replaces gpt-5 with $+5.69$ points, $\ell = +0.16$ points, Holm-adjusted exact $p = .043$, at cost ratio 0.0037 in USD; the ArenaHard creative-writing cell ($+11.46$ points, $n = 48$) misses the Holm threshold ($p = .065$), and the ArenaHard math and MMLU-Pro cells fail and keep the large model. The never-opened confirmation role, authorized only for that cell and those two models, confirms it: 54.84% against 47.71% on 1,291 requests, $+7.13$ points and 13.63% relative error reduction, both halves positive ($+5.46$/$+8.86$), $\ell = +2.83$ points, paired bootstrap interval $[3.80, 10.30]$ points, exact $p = 2.9 \cdot 10^{-5}$, average USD cost ratio 0.0036 with zero violations, so both the improvement gate and the 30%-saving-within-one-point gate pass. This is a one-action root result in a second, larger-scale pool that shares no model with the first, under a second cost unit; the three cells that failed at validation are retained as frozen negatives. The official AvengersPro router, trained on the same fit role with its pinned 7B embedder and evaluated on the mechanism role under the identical USD budget (its embedding call charged zero, as for every router forward pass in Table 9; at every input price that the pool publishes, charging the call gives 9 cap violations and 0.901 to 0.915 times the average USD cost of gpt-5, Table 10), reaches 74.20% macro accuracy at cost ratio 0.90 with one cap violation replaced by gpt-5; the fit-selected root policy reaches 73.07% at cost ratio 0.68 on the same rows, a paired difference of -0.24 points with bootstrap interval $[-1.60, 1.11]$, and the mechanism-gated root policy 74.75% at cost ratio 0.78. The router is more accurate on ArenaHard coding ($n = 37$) and worse on ArenaHard math and MMLU-Pro; none of its cells passes a gate.

RCPSP planning: validation only. The planning pools reached validation and certify nothing. On RCPSP Pool A the policy that runs Qwen2.5-Coder-1.5B and then LFM produces 311 rescues and 124 harms over its same-family counterpart, and stopping at the earliest certificate adds 50 rescues and no harms over the terminal LFM response; nevertheless the earlier gate fails, because the relative error reduction over the two 3B models is 3.50 to 3.58%, below the 5% threshold. Pool B replaces both families by Falcon-H1-1.5B and Granite-1B; it yields 153 rescues and no harms over the same-family control, and removing Granite-1B recovers that control exactly; its authorization permits validation, not confirmation. Useful stopping does not imply useful feedback-conditioned generation: in the $n = 1{,}000$ RCPSP pilot, feeding verifier feedback into the certifier loses 2.30 points (nine rescues, 32 harms) relative to the certifier run without feedback.

E.4 Failure modes

Rare acceptance: ProofNet. The proof-checker cell was registered on ProofNet, whose Lean 4 statements were checked in the same isolated toolchain (364 of 371 compile; 182 valid statements form the mechanism role in two halves, and 182 test statements are reserved as a confirmation role), with the gate copied verbatim from the MiniF2F cell. The mechanism gate fails and the cell is frozen: the policy certifies $35/182$ against $14/182$ for Goedel-Prover-V2-32B, the large model (21 rescues, 0 harms; halves $14/7$ and $21/7$; $\ell = +4.77$ points; exact $p < 10^{-6}$) with zero violations, but at cost ratio 0.888 (0.903/0.875; bootstrap $[0.846, 0.925]$), above the 0.70 ceiling, and the 7B certifier's acceptance rate of 18.1% ($33/182$) triggers the registered rule that blocks the cell. The cost

identity of Theorem 1 accounts for the ratio exactly ($\alpha = 0.020$, $q_1 = 0.181$, $\kappa = 1.10$, $\tau = 0.964$): the certifier rarely stops the request, so the large model runs on 82% of the statements and its own sample exceeds the 7,800-token prefix on 108 of them. The fixed models expose a second limit: the 7B prover alone certifies $23/182$ at the 8,192-token cap, more than the large model registered from MiniF2F, so the improvement gate's error condition against the strongest fixed model (7.5% relative error reduction) would fail as well. ProofNet-test was never opened. Three fresh-seed repeats of the mechanism cell (same statements, caps and prompts) reproduce both sides of the result: cost ratios 0.897, 0.880 and 0.866 against the sealed 0.888, net gains of $+20$, $+26$ and $+29$ statements against $+21$, and zero harms in every draw; the cost condition fails on every draw and the accuracy gain is systematic.

Ceiling failures at the 32B large model. The third turn of the five-language Multi-IF series (723 mechanism rows, three cumulative constraint sets) is the boundary of that series: the policy is still ahead, $398/723$ against $367/723$ ($+4.29$ points; 39 rescues, 8 harms; $\ell = +1.5$), but the certifier's acceptance rate falls to 49.7% and the cost ratio rises to 0.676 with one half at 0.715, above the 0.70 ceiling, so the registered gate fails and the cell is frozen with its confirmation role closed. Across the three turns the acceptance rate falls from 0.77 to 0.59 to 0.50 and the realized cost ratio rises from 0.43 to 0.58 to 0.68, as the identity predicts. IFBench's out-of-distribution subset is the boundary of the IFBench cells: on its mechanism role the policy is ahead, $205/409$ against $184/409$ ($+5.1$ points; 21 rescues, no harm; $\ell = +2.1$), but the certifier's acceptance rate is only 42.5% and the cost ratio is 0.712 with one half at 0.729, above the 0.70 ceiling, so the registered gate fails and the cell is frozen with its confirmation role closed, exactly the outcome that its authorization had named as the risk. Both cells are served as the large model alone; the 72B results on the same requests are given above.

An inexact certificate: LiveCodeBench. On LiveCodeBench the public tests certify an answer while the hidden tests decide correctness. Across 3,060 frozen candidates on 85 official tasks, 249 candidates pass the public examples but only 162 also pass the hidden evaluation: the public-pass precision is 65.06% and the false-discovery fraction among public passes is 34.94%. Fifteen tasks have at least one false public pass but no true public pass, and six contain both kinds. The rates vary by model: the public-pass precision is 82.76% for LFM-1.2B, 68.57% for Qwen-1.5B and 54.55% for Qwen-3B, so the larger model creates more hidden-correct candidates (66 against 48 for either small model) but a less exact certificate. This post-label diagnostic opened no new prompts or candidates; the 3,060 candidate slots correspond to 1,375 unique executions, so slots are not treated as independent units. The harm bound of Proposition 5 on this benchmark is tested in Table 17, Appendix F.

E.5 Cost predictions

Predictions registered before the large model ran. Every 72B cost ratio was registered from the corresponding 32B cell through Proposition 2 before the 72B ran: the realized P2 cost ratio differs from the registered one by at most 0.011 on the five turn-1 and structured-output cells, and the P3 realized ratios are at or below their registered values except on English, French and Spanish turn 3, where the registered failure did not occur. On the pooled development rows of the 32B cells, a predictor of each request's conditional cost beats prediction from the rejection rate alone, in both fold directions. The cross-family ladder and its registered band are in Table 18, Appendix F, and the certifier ablation in Table 20.

E.6 Routers and cascades

Routers and cascades under the same budget. Table 9 places the standard alternatives under the identical per-request cap plus average budget on the development roles of both official settings, each trained on the fit role: an embedding router with per-model logistic predictors (quality-first, and a cost-aware variant that picks the cheapest model predicted no worse than the large model), a FrugalGPT-style cascade whose accept rule is the learned answer-value reliability and whose model sequence has at most two models, majority voting over the best cap-feasible subset of at most three models, and the trained AvengersPro router. On the Performance-Cost pool every arm of Table 9 except random routing and majority voting lies within two macro points of the 73.07% macro ac-

curacy of our fit-selected root, and cost separates the arms. Random routing (cost ratio 0.318), cascade-routing (0.587), our fit-selected root (0.682), the cost-aware embedding router (0.727), the cost-aware matrix-factorization router (0.729) and our mechanism-gated root (0.777) cost less than 0.8 times the average cost of gpt-5. The quality-first matrix-factorization router spends 0.888, the quality-first embedding router spends 0.896, AvengersPro spends 0.901, the reliability cascade exceeds the large model's spend (1.05), and majority voting breaks the per-request cap on 114 of 2,475 requests. Per request, the paired lower bounds of our fit-selected and mechanism-gated root policies are -1.74 and -1.01 points, and those of the reliability cascade and majority voting are $+5.20$ and $+2.74$ points. The released prediction of SimpleQA holds the grader's letter rather than the model's answer, and the reliability cascade and majority voting treat that letter as an answer. When the large model answers every SimpleQA request, the exact lower bound of the reliability cascade is -1.11 points and the exact lower bound of majority voting is -3.86 points. On the Performance-20 pool cascade-routing is more accurate than the large model alone in macro accuracy (70.54 against 67.06) and in per-request accuracy ($+3.63$ points), at cost ratio 0.851 with zero cap violations. The cost-aware matrix-factorization router gains 0.18 macro points and loses 0.18 points per request at cost ratio 0.881. The quality-first embedding and matrix-factorization routers gain 2.47 and 3.18 macro points at cost ratios 1.178 and 1.165 with two cap violations each, and the reliability cascade and majority voting cost 1.590 and 1.740 times the large model. Our root policies equal the large model alone in macro accuracy and in cost on this pool, and no Performance-20 context passes the mechanism gate of Table 16. The paired lower bounds of the router baselines, here and in Section 4, split $\delta = 0.05$ over the nonzero paired differences that occur on each arm. On both pools a reward takes a value in $\{0, \frac{1}{2}, 1\}$, so a paired difference takes a value in $\{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$. Splitting δ over that support, as Definition 2 splits it over $\{-1, 0, +1\}$, gives -1.86 and -1.11 points for our fit-selected and mechanism-gated root policies on the Performance-Cost pool, $+4.76$ and $+2.25$ points for the reliability cascade and majority voting on that pool, and $+0.33$ points for cascade-routing on the Performance-20 pool; each of these bounds has the sign of the bound quoted for the same arm. The bound of the reliability cascade on the Performance-20 pool, which the paper does not quote, changes from $+0.30$ to -0.19 points. Random routing is 10.80 macro points below our fit-selected root on the Performance-Cost pool and 18.68 macro points below it on the Performance-20 pool. On the Performance-Cost pool the best single model is gpt-5, the large model itself, whether it is chosen on the fit role or in hindsight. On the Performance-20 pool the best single model chosen on the fit role is GLM-Z1-9B-0414, which differs from the large model only on MBPP; it is 0.67 macro points below the large model at cost ratio 1.048 with 34 cap violations. The best single model chosen in hindsight on that pool is Qwen3-8B, 1.52 macro points above the large model at cost ratio 1.119 with 17 cap violations.

E.7 What the record does not establish

Confirmed claims and everything else. The confirmed numbers are those of the eight 32B cells and their three-seed repeats, the nine 72B cells, the 128-task MiniF2F cell, the SimpleQA cell on Performance-Cost, the MBPP context on Performance-20, and the IFEval-es, M-IFEval and HEADLINES cells under their earlier gates, each on its own requests under the gate that its authorization named (Section 3). Every other number in this appendix is a development result, a frozen negative or a post-label diagnostic, and carries no claim. The RCPSP pools reached validation only. The refinement tree on Performance-20 selected no node over the large model, and the MBPP context is the only context on which the root rule changes the action. The hybrid policy that served verifiable requests by certificate stopping and the rest by a learned router at the unspent budget failed its gate (Appendix F). The MiniF2F-test cell is a development row. The frozen negatives, ProofNet, Multi-IF turn 3 in five languages at both large models, IFBench's out-of-distribution subset at the 32B large model and under P3 at the 72B, and the three Performance-Cost validation cells, are never retried and are served as the large model alone. Nothing in this record shows that composition beats the strongest available model on tasks that no verifier covers.

F Evidence tables

This appendix holds the tables and the descriptive comparisons that Section 4 and Appendix I cite, in the order in which Section 4 cites them. Every number is exact arithmetic over recorded per-request outcomes and costs: no table here runs a model, opens a role or changes a verdict. The Qwen3-32B

Table 9: Routing and cascade baselines under the identical per-request-cap plus aggregate contract on the development roles of the two official LLMRouterBench settings (Performance-Cost 13-model API pool in USD, mechanism role; Performance-20 open-checkpoint pool in parameter-weighted tokens, validation role). Macro accuracy after the response-safe wrapper (an over-cap decision falls back to the protected answer and is counted as a violation), aggregate cost ratio to the protected comparator, and raw cap violations. The protected comparator is gpt-5 on Performance-Cost, and GLM-Z1-9B-0414 on Performance-20, except Qwen2.5-Coder-7B-Instruct on MBPP. The same charging rule holds on both pools: every router forward pass is charged zero. This rule covers the 7B embedding call that AvengersPro makes, and it covers the same call that the embedding, matrix-factorization, cascade-routing and AutoMix arms make; Table 10 reports these arms with the call charged. AvengersPro is scored from its own released decisions. On Performance-20 AvengersPro was trained on the fit and mechanism roles, while every other trained arm was trained on the fit role only. The best single model is one model deployed on every request of the pool, selected by macro accuracy on the fit role or, in hindsight, by macro accuracy on the evaluation role: gpt-5 is selected by both rules on Performance-Cost, and GLM-Z1-9B-0414 is selected on the fit role and Qwen3-8B is selected in hindsight on Performance-20. Random routing is the exact expectation of a uniform choice, made before any cost is paid, among the models whose cost fits the request budget on every fit-role request of the same dataset; its violations are expected counts. No baseline cell carries a certificate.

	Performance-Cost (USD)			Performance-20 (PWT)		
policy	macro	cost	viol.	macro	cost	viol.
Protected comparator alone	71.87	1.000	0	67.06	1.000	0
Best single model, chosen on the fit role	71.87	1.000	0	66.39	1.048	34
Best single model, chosen in hindsight	71.87	1.000	0	68.58	1.119	17
Random routing over cap-admissible models (expectation)	62.27	0.318	0.31	48.38	0.714	0.14
StateCompose root (fit-selected)	73.07	0.682	0	67.06	1.000	0
StateCompose root (mechanism-gated)	74.75	0.777	0	67.06	1.000	0
AvengersPro (trained embedding router)	74.20	0.901	1	68.27	1.203	0
Embedding LR router, quality-first	72.61	0.896	2	69.53	1.178	2
Embedding LR router, cost-aware	73.26	0.727	0	66.95	0.872	0
Matrix-factorization router (RouteLLM-style), quality-first	73.06	0.888	2	70.24	1.165	2
Matrix-factorization router (RouteLLM-style), cost-aware	73.47	0.729	0	67.24	0.881	0
FrugalGPT-style reliability cascade	72.07	1.052	0	67.42	1.590	0
Majority vote (cap-feasible subset)	60.41	1.394	114	53.93	1.740	79
Cascade-routing (Dekoninck et al., 2025)	71.23	0.587	0	70.54	0.851	0
AutoMix-style self-verification cascade	71.41	0.924	7	57.51	0.847	41

cells, the ProofNet cell and the eleven Qwen2.5-72B cells are listed in Table 6 of Appendix E and are not repeated here.

Cells confirmed under their own earlier gates, and cells that reached validation only. The table below lists the cells sealed before the efficiency gate of Definition 2 existed, the MiniF2F-test development cell and the planning cells. RER is the relative error reduction: the error rate of the stated comparison model minus the error rate of the policy, divided by the error rate of the stated comparison model. Costs are in parameter-weighted tokens unless USD is stated.

Task and cell	Public state and role	Outcome and claim boundary	Status
Instruction following (IFEval, Spanish)	Unmet-constraint set of the verifier; confirmation role never opened before	87.14% against 78.88% for Qwen3-4B and for the strongest registered same-family composition; $+8.25$ points, 39.08% RER; every per-request and average cost gate, in parameter-weighted tokens and in floating-point operations, passes.	Confirmed under its own earlier gate
Instruction following (M-IFEval, French and Japanese)	The same verifier state; confirmation role never opened before	78.62% against 71.01% for the external large model and 71.50% for the deployable same-family composition; $+7.62$ points and 26.27% RER over the external large model; positive on every language and on every half; the average cost is 20.36% lower in parameter-weighted tokens and 22.34% lower in floating-point operations. This is a replication of one application class, not a new class.	Confirmed under its own earlier gate

Task and cell	Public state and role	Outcome and claim boundary	Status
Finite-label classification (HEADLINES)	Factorized evidence state; official test set never opened before	79.42% against 72.62% for the best single member and 64.20% for the external 3B large model; without the cross-family member the policy reaches 77.64%, so that member adds $+1.78$ points with positive halves; the cost is 20.06% below the 3B large model's. Accuracy is equivalent, not superior, to logistic stacking (79.80%) and to the Chow–Liu Bellman policy (79.50%).	Confirmed under its own earlier gate
Natural-language inference (SNLI)	Shared-choice evidence state; official test set never opened before	80.61% against 77.85% for Qwen3-8B chat; $+2.76$ points and 12.45% RER; $+0.62$ points above the Qwen3-8B self-composition control with the identical mask, at 9.57 to 9.70% less cost; above single-pass Qwen3-14B but below the costlier 14B chat-to-raw cascade.	Confirmed under its own composition-superiority gate
Formal theorem proving (MiniF2F, 128 tasks)	Exact Lean 4.16 certificate; confirmation role never opened before	$91/128$ against $85/128$ for the strongest fixed code-first action; six rescues, no harms, 13.95% RER, positive halves, Holm-adjusted $p = 0.03125$; the largest per-request cost ratio is 0.891 and the average cost ratio 0.443.	Confirmed under its own earlier gate
Planning (RCPSP, Pool A)	Exact schedule certificate; independent role	7.93% against 4.59% for the large model and 6.06% for the same-family composition; the effect is positive, but the RER is 3.50 to 3.58%, below the frozen 5% gate.	Mechanism role only
Planning (RCPSP, Pool B)	Exact schedule certificate; prospective validation role	6.00% against 4.52% for the large model and 4.47% for the same-family composition; 153 rescues and no harms, on the validation role and not on a confirmation role.	Validation role passed; not confirmed
Code (LLM-RouterBench MBPP, Performance-20)	Public benchmark identity as the context; confirmation role never opened before	The frozen Qwen2.5-Coder-7B action reaches 77.72% against 64.92% for GLM-Z1-9B: $+12.79$ points, 36.45% RER, positive halves, average cost ratio 0.0553. Over all ten contexts the policy gains $+1.62$ points, confidence interval $[0.87, 2.37]$ points, with a largest per-request cost ratio of one. The official AvengersPro is more accurate only before per-request cost admission: its action sequence after the response-safe rule is identical, while its 7B embedding call makes the fallback requests infeasible under the full cost accounting.	Confirmed
Code (LLM-RouterBench, the remaining eight Performance-20 contexts)	Public benchmark identity as the context; validation role	The same root rule with a Qwen2.5-Coder-7B action on HumanEval reaches $845/1{,}269$ against $842/1{,}269$ for fixed Qwen3-8B over the eight contexts, and $21/29$ against $18/29$ on HumanEval alone. The paired gain is not significant (Holm-adjusted $p = 1$), so validation fails closed and the confirmation role stays closed.	Failed closed
Frontier API models (LLMRouter-Bench Performance-Cost, 13 models, USD prices)	Public benchmark identity as the context; a second pool at a second scale	The registered root rule with complete cascade actions under the per-request cap and the average budget, against gpt-5. Four cells pass the mechanism gate; validation keeps only SimpleQA ($+5.69$ points, exact lower bound $+0.16$ points, Holm $p = 0.043$, cost ratio 0.0037 in USD). On the never-opened confirmation role: 54.84% against 47.71% on 1,291 requests, $+7.13$ points, 13.63% RER, halves $+5.46$ and $+8.86$, exact lower bound $+2.83$ points, confidence interval $[3.80, 10.30]$ points, exact $p = 2.9 \cdot 10^{-5}$, cost ratio 0.0036 in USD, zero cap violations. ArenaHard creative writing ($+11.46$ points, $n = 48$, Holm $p = 0.065$), ArenaHard math and MMLU-Pro (cascade at cost ratio 0.536, -0.16 points) fail closed at validation and are frozen negatives. The official AvengersPro on the mechanism role under the same USD accounting: 74.20% macro accuracy at cost ratio 0.90 (one violation, handled by the response-safe rule) against the fit-selected root policy's 73.07% at 0.68, paired -0.24 points, confidence interval $[-1.60, 1.11]$ points.	Confirmed (SimpleQA); three frozen negatives

Task and cell	Public state and role	Outcome and claim boundary	Status
Formal theorem proving (MiniF2F-test, 244 tasks; second scale)	Exact Lean certificate; a frozen 2,048-token pilot on 16 opened valid tasks, a cap calibration on 8 opened valid tasks, the mechanism role on 104 opened valid tasks, and the confirmation role on MiniF2F-test (244 tasks, never opened)	The frozen pilot fails closed (the 32B proves $7/16$ code-first and $2/16$ native, with zero rescues over the 7B); at a calibrated cap of 8,192 tokens the 32B certifies $7/8$. Efficiency cell with zero fitted parameters: the 7B code-first prover, then the certificate, then the large model's prefix (7,800 tokens) under the large model's cap. Mechanism role: 80 against 75 of 104 at cost ratio 0.533, zero harms. Confirmation role: 154 against 142 of 244, $+4.92$ points, exact lower bound $+0.24$ points, confidence interval $[+2.05, +8.20]$ points, $p = 0.0042$, cost ratio 0.692, halves 0.679 and 0.704, zero violations. Neither gate is awarded: the per-half ceiling of the efficiency gate is missed at 0.704, and the improvement bound changes sign with the split of δ over the support (positive over two values, negative over the three-way split used everywhere else).	Develop-ment

Reference points on the eleven cells with a second large model. Table 13 gives, for the eleven cells on which the frozen policy faced Qwen3-32B and then Qwen2.5-72B, the accuracy and cost of the certifier alone, of the large model, of the policy, of random routing, of a matched-call random escalation and of the two-action hindsight oracle; all eleven are mechanism-role measurements. With an exact verifier the policy attains 0.87 to 1.00 of the hindsight oracle's gain over random routing, and the difference is made only of certifications lost to truncation and of the certifier's cost on rejected requests.

The best-of-k sampling control. Table 14 runs the control of Section 4 on the mechanism roles: the certifier is sampled up to k times under the cell's per-request budget, the first answer that the verifier accepts is delivered, every sample is charged, and k is the largest value whose total cost does not exceed the policy's on the same requests. The table matches the control to the policy on average cost only, so Table 15 reports whether the control also respects the per-request budget, from a replay that charges samples exactly as the table does and reads each request's budget from the policy's own recorded outcomes. At the budget-matched k the control exceeds the per-request budget on 38 of the 726 Multi-IF five-language requests, on 5 of the 523 IFBench classic requests, on 2 of the 415 JSONSchemaBench GitHub requests and on none of the 309 JSONSchemaBench requests from other collections; the first request over budget appears at $k = 19$ on Multi-IF and on IFBench and at $k = 24$ on JSONSchemaBench GitHub. The policy exceeds the per-request budget on none of the requests of the four cells. Where the control matches the policy in average cost, it therefore does not also meet the bound on every request. MiniF2F uses a different replay and is outside this measurement; its absent entry is not a zero.

The 21 routing contexts without a verifier. Table 16 consolidates the fit-to-mechanism diagnoses of the Performance-20 and Performance-Cost settings, 21 public contexts, under the per-request cap and the average budget. Each context is assigned one loss class by fixed rules: certified (the registered root rule passed the mechanism gate), cost (a cascade improves only above the cost-ratio ceiling), identifiability (a positive fit-role gain that does not survive the two disjoint mechanism halves), no headroom (the oracle union is within one point of the large model), or uncertified. Four contexts passed the mechanism gate, all in the Performance-Cost pool; five are cost-bound and three identifiability-bound. On Performance-20, replacing GLM-Z1-9B by the model selected on the fit role changes mechanism-role accuracy by -14.9 to $+4.4$ points, and delivering each request's best answer in the pool gains $+3.4$ to $+48.7$ points over GLM-Z1-9B. The SimpleQA cost ratio displayed as 0.00 in the table is a rounded entry, not zero cost: its stored value is 0.003511437164480004 on the 860 mechanism-role requests, relative to gpt-5's average cost on those same requests, and it is not the ratio of the separate confirmation role.

The harm bound under an inexact certificate. On LiveCodeBench the public tests certify while the hidden tests decide, so the certificate is strictly weaker than the metric. Table 17 tests the bound

Table 10: The embedding call charged, for AvengersPro and for the embedding, matrix-factorization, cascade-routing and AutoMix arms, each of which computes a 7B GTE-Qwen2 embedding of the request. Columns headed zero repeat Table 9, where the call is charged nothing on both pools. Columns headed charged add the cost of the call on every request on which the arm computes it, including requests whose decision the wrapper replaces. In parameter-weighted tokens the charge is 7 billion parameters times the input tokens that the embedder reads, at most 7,500 per request (token-count. In USD the benchmark publishes no price for the embedder, so the charged column uses the lowest input price that the pool publishes for any model, 0.09 USD per million tokens (qwen3-235b-a22b-2507). At every input price that the pool publishes, from 0.09 to 3 USD per million tokens, AvengersPro records 9 cap violations and a cost ratio between 0.901 and 0.915. On 8 requests the cost of the AvengersPro decision already equals the request budget, and that budget equals the realized cost of gpt-5 itself, so any positive charge exceeds the budget on those requests even after the fallback. Under the request-wise budget of the Performance-20 code cell, where the budget of a request is the realized cost of GLM-Z1-9B-0414 on that request, AvengersPro exceeds the budget on 861 requests with the call charged zero and on 1,097 requests with the call charged.

	Performance-Cost (USD)						Performance-20 (PWT)					
	zero			charged			zero			charged		
policy	macro	cost	viol.	macro	cost	viol.	macro	cost	viol.	macro	cost	viol.
AvengersPro (trained embedding router)	74.20	0.901	1	74.20	0.901	9	68.27	1.203	0	68.27	1.228	0
Embedding LR router, quality-first	72.61	0.896	2	72.61	0.897	10	69.53	1.178	2	69.53	1.203	2
Embedding LR router, cost-aware	73.26	0.727	0	73.26	0.727	8	66.95	0.872	0	66.95	0.897	0
Matrix-factorization router (RouteLLM-style), quality-first	73.06	0.888	2	73.06	0.889	10	70.24	1.165	2	70.24	1.191	2
Matrix-factorization router (RouteLLM-style), cost-aware	73.47	0.729	0	73.47	0.729	8	67.24	0.881	0	67.24	0.907	0
Cascade-routing (Dekoninck et al., 2025)	71.23	0.587	0	71.23	0.587	0	70.54	0.851	0	70.54	0.877	0
AutoMix-style self-verification cascade	71.41	0.924	7	71.41	0.925	7	57.51	0.847	41	57.51	0.872	51

Table 13: Reference points on the eleven second-comparator cells, computed exactly from the sealed per-request outcomes and costs with no further model calls. *Random escalation* pays the certifier on every request and then escalates a uniformly random subset of the policy's own size, so it is matched to the policy in calls but not in which calls. Accuracies are the official checker's; costs are parameter-weighted tokens as a fraction of the comparator's own. *Oracle* delivers, on each request, the certifier's answer when the verifier accepts it and otherwise the comparator's full-length answer, and pays for that one action only; it is a hindsight reference that no deployable policy can attain. The last column is the policy's gain over the *random-routing* line (the straight line between the two single-model points) at the policy's cost, divided by the oracle's gain over the same line at the oracle's cost. With an exact verifier this share falls below 1 only through certifications lost to truncation and through the certifier's cost on rejected requests less the truncation saving; it does not measure selection between the two actions.

cell	n	comparator acc.	certifier acc. @ cost	policy acc. @ cost	rand. esc. acc.	oracle acc. @ cost	realized share
Multi-IF 5 languages, turn 1	726	78.2	77.4 @ 0.061	**87.3** @ 0.345	79.7	87.3 @ 0.332	1.00
Multi-IF 5 languages, turn 2	726	59.9	61.4 @ 0.055	**68.6** @ 0.503	64.4	69.0 @ 0.488	0.95
Multi-IF 5 languages, turn 3	723	42.6	49.5 @ 0.055	**54.2** @ 0.628	52.1	54.6 @ 0.611	0.97
Multi-IF En/Fr/Es, turn 2	567	69.3	74.3 @ 0.055	**78.5** @ 0.322	75.4	78.7 @ 0.310	0.98
Multi-IF En/Fr/Es, turn 3	554	54.9	61.7 @ 0.055	**66.4** @ 0.489	64.0	67.7 @ 0.474	0.87
Multi-IF English, turn 1	258	84.9	83.3 @ 0.061	**89.5** @ 0.198	84.4	89.5 @ 0.189	1.00
Multi-IF Fr/Es, turn 1	309	87.1	81.9 @ 0.058	**91.9** @ 0.233	83.8	92.6 @ 0.223	0.93
IFBench classic (WildChat)	523	83.2	79.0 @ 0.054	**87.6** @ 0.284	80.8	87.6 @ 0.273	0.99
IFBench out-of-distribution	409	51.8	42.5 @ 0.057	**58.7** @ 0.651	51.8	58.7 @ 0.617	0.97
JSONSchemaBench GitHub	415	88.7	81.0 @ 0.056	**91.8** @ 0.327	83.0	91.8 @ 0.312	0.99
JSONSchemaBench other collections	309	93.5	90.6 @ 0.055	**95.1** @ 0.188	91.0	95.1 @ 0.183	1.00

The verifier on these cells is exact, so on every request the certifier accepts the policy and the oracle deliver the same certified answer. The realized share, between 0.87 and 1.00, therefore reflects two losses and no selection. The accuracy loss is 16 rejected requests, summed over the 11 cells, on which the residual cap truncated a comparator answer that is certified at full length; no truncated answer is certified where the full-length answer is not. The policy's accuracy therefore equals the oracle's on 6 of 11 cells and is at most 1.26 points below it on the others. The cost loss is the certifier's cost on rejected requests less the truncation saving, 0.006 to 0.034 of the comparator's cost. Where the certifier is more accurate than the comparator (Multi-IF 5 languages, turn 2; Multi-IF 5 languages, turn 3; Multi-IF En/Fr/Es, turn 2; Multi-IF En/Fr/Es, turn 3), the random-routing line falls with cost, so the cost loss raises the share instead of lowering it.

$q_a \cdot (1 - p_a)$ of Proposition 5 overall and on each difficulty stratum of the pilot's already-open fit role, with Qwen3-32B as the large model; the pilot is not reopened and no certificate is awarded. The slack of the bound widens with difficulty, from 1.00 of the bound on easy problems to 0.14 on hard.

Table 14: The budget-matched control: what repeated sampling of the certifier buys at the same budget. For each cell the certifier is sampled k times under the cell's own per-request cap and the first output the official verifier accepts is delivered, charging *every* sample drawn. The control column reports the largest k whose total cost does not exceed the cascade's own cost on the same requests; every column of a row is taken at that single k. Paired differences and exact one-sided bounds are cascade minus control, request by request.

cell	n	cascade acc. @ cost	best budget-matched control acc. @ cost	k	paired (pt)	exact LCB
Multi-IF, 5 languages, turn 1	726	87.33 @ 0.345	86.91 @ 0.341	24	+0.41	-0.028
IFBench classic	523	87.57 @ 0.284	83.56 @ 0.279	22	+4.02	+0.007
JSONSchemaBench GitHub	415	91.81 @ 0.327	86.75 @ 0.323	24	+5.06	+0.009
JSONSchemaBench other	309	95.15 @ 0.188	93.85 @ 0.167	24	+1.29	-0.027
MiniF2F (Lean compiler)	104	76.92 @ 0.533	78.85 @ 0.193	16	-1.92	n/a

The cascade certifies more than the best budget-matched control on 4 of the 5 cells, significantly on 2. On MiniF2F it does not: the Lean control is cheaper at every k drawn and *dominates*, matching the cascade's certified fraction at $k{=}4$ for $0.071\times$ the comparator against the cascade's $0.533\times$, and exceeding it by $k{=}7$. The identity explains the split: the Lean certifier costs $\alpha \approx 0.034$ per sample, so sixteen draws still cost under a fifth of the comparator, whereas the instruction-following certifiers are relatively far more expensive. Whether escalation or repeated sampling wins is therefore decided in advance by α, not by which is intrinsically better.

Mechanism subset	Requests	Best-of-k violations	Policy violations
Multi-IF, five languages	726	38	0
IFBench, classic	523	5	0
JSONSchemaBench, GitHub	415	2	0
JSONSchemaBench, other	309	0	0
MiniF2F	—	not measured	not measured

Table 15: Per-request budget violations accompanying Table 14. Counts concern the budget-matched k on the stated mechanism subsets. MiniF2F is outside this replay; its absent measurement is not a zero.

Cost ratios predicted before the large model ran. Table 18 places, on the 24 cells of the cross-family ladder, phi-4, Mistral-Small-24B, Yi-1.5-34B and Nemotron-70B on six subsets, the realized cost ratio against the ratio that Proposition 2 predicted from an earlier run of the same certifier, with λ the parameter ratio, before any token of the new large model existed. The registration named 0.05 as the falsifier in advance. These cells have a mechanism role and no confirmation role, so a passing gate there adds no certified claim. Of the 24 cells, 21 pass their gate with no cap violation; the three failures are on IFBench's OOD subset.

The admission condition, in sample and out of sample. Admitting the 32B as a middle stage follows Theorem 4. On the eleven cells that ran both the two-stage and the three-stage policy, the classical index is right on nine and the condition of Theorem 4, which contains it, on all eleven; that condition is a fit derived after those cells were sealed (Table 4, Appendix C). The cross-family ladder was registered as the out-of-sample test of the condition before that ladder ran. There the condition is right on 24 of 24 cells against 21 of 24 for the classical index, disagreeing on three and right on each (Table 19).

Cap violations of the three-stage policy. Table 19 reports the three-stage policy's realized cost, and two facts about that policy belong beside it. On 16 of the 24 cells it exceeds the per-request budget $B(x) = C_P(x)$ on at least one request, 872 requests in total; the worst cell is IFBench OOD against phi-4, which exceeds the budget on 244 of its 695 requests (35.1%), because the sealed rule runs each certifier stage in full and the 32B middle stage alone can cost more than the entire budget of a large model smaller than the 32B. A zero-violation requirement is part of the mechanism gate, so the three-stage policy fails its gate on those cells and they are frozen negatives; the gate passes on exactly the seven cells where the middle stage paid. On a violating request no prefix of the large model fits, so the large model contributes no recorded cost and the printed three-stage ratios for

Table 16: Where composition value is lost on the two official LLMRouterBench settings (P20 = Performance-20, PC13 = Performance-Cost; fit $\rightarrow$ mechanism, cap plus aggregate contract). Gains in points versus the protected comparator (mechanism halves in parentheses); loss class by the fixed rules of the artifact.

pool	context	n	fit action	gain (halves)	ratio	class
P20	arenahard/test	148	DeepSeek-R1-0528-Qwen3-8B	+1.0 (+2.3/-0.8)	0.85	identifiability
P20	bbh/test	216	NVIDIA-Nemotron-Nano-9B-v2	+0.9 (+1.0/+0.9)	0.58	uncertified
P20	finqa/test	216	NVIDIA-Nemotron-Nano-9B-v2	+1.4 (+2.8/+0.0)	0.93	cost
P20	gpqa/test	39	internlm3-8b-instruct	-12.8 (-9.1/-17.6)	0.27	cost
P20	livecodebench/test	218	Llama-3.1-Nemotron-Nano-8B-v1	-11.0 (-9.7/-12.4)	0.49	uncertified
P20	livemathbench/test	30	DeepSeek-R1-Distill-Qwen-7B	-3.3 (-5.9/+0.0)	1.17	cost
P20	math500/test	87	Fin-R1	-14.9 (+2.5/-29.8)	0.35	uncertified
P20	mbpp/test	185	internlm3-8b-instruct	-15.1 (-20.0/-10.5)	0.49	uncertified
P20	mmlupro/test_1000	229	NVIDIA-Nemotron-Nano-9B-v2	+3.1 (-2.1/+6.8)	0.88	cost
P20	winogrande/valid	227	Qwen3-8B	+4.4 (+10.2/-0.8)	0.70	cost
PC13	aime/hybrid	9	glm-4.6	-11.1 (+0.0/-25.0)	0.62	no headroom
PC13	arenahard/test	148	qwen3-235b-a22b-2507	-1.7 (-4.0/+1.6)	0.03	identifiability
PC13	arenahard_coding/test	37	gpt-5-chat	-17.6 (-35.4/+15.4)	0.25	identifiability
PC13	arenahard_creative_writing/test	60	deepseek-r1-0528	+13.3 (+8.3/+20.8)	0.16	certified
PC13	arenahard_math/test	44	gpt-5-chat	+14.8 (+25.0/+6.2)	0.24	certified
PC13	gpqa/test	39	deepseek-r1-0528	-17.9 (-22.7/-11.8)	0.44	uncertified
PC13	hle/test	442	glm-4.6	-10.6 (-12.4/-8.9)	0.58	uncertified
PC13	livecodebench/test	218	deepseek-r1-0528	-3.7 (-7.1/+0.0)	0.55	uncertified
PC13	livemathbench/test	30	deepseek-r1-0528	+3.3 (+5.9/+0.0)	1.18	uncertified
PC13	mmlupro/test_3000	588	deepseek-r1-0528	-5.1 (-5.2/-5.0)	0.65	certified
PC13	simpleqa/test	860	qwen3-235b-a22b-2507	+2.6 (+4.6/+0.2)	0.00	certified

those cells understate what the policy would spend. The two-stage policy of Table 18 is unaffected: its violations are zero on all 24 cells.

Moving the certifier. Everywhere else in this paper the acceptance rate is observed rather than set: one certifier serves all eight confirmed cells and all 24 cross-family cells, so the acceptance rate cannot vary there by construction. Table 20 varies it directly on the Multi-IF turn-1 cell, holding the large model's recorded outputs, the roles, the generation settings and the checker fixed and changing only the certifier. It is a mechanism-role measurement and carries no certificate. Four predictions were registered; two hold and two fail. The acceptance rate rises with certifier size at every step, and almost all of the rise is bought early: 31.68 of the 33.47 points by the 4B, and the last factor 3.7 in parameters buys 1.79. The realized cost is not monotone in size, and its minimum falls at the interior point registered in advance, because the 14B is accepted barely more often while costing three times as much to run. The transfer assumption fails: carrying $\kappa \cdot \tau$ from the 4B cell over-predicts every other certifier, by up to 0.1073 against a registered falsifier of 0.05, in the same direction each time. The identity itself cannot fail; what fails is treating $\kappa \cdot \tau$ as a property of the cell when it belongs to the certifier and the large model jointly: it falls as the certifier weakens and its rejected set grows larger and therefore less selected, and it falls again as the certifier grows expensive and leaves less

Table 17: The harm bound where the certificate is strictly weaker than the metric. On LiveCodeBench the public example tests certify and the private tests decide, so a certificate can be wrong; the benchmark's own difficulty label splits the rows into strata whose precision π ranges from 0.36 to 0.97. Proposition 5 bounds the accuracy a certificate-first policy can lose by $q(1-\pi)$; the realized harm is what it would actually have lost, which requires knowing when the comparator would have been right and is therefore measured here rather than assumed. The comparator is Qwen3-32B on the pilot's already-open fit role; the blocked pilot is not reopened and no certificate is awarded.

stratum	n	certified	q	π	bound $q(1-\pi)$	realized
all problems	126	57	0.452	0.807	0.0873	0.0317
easy	33	29	0.879	0.966	0.0303	0.0303
medium	38	17	0.447	0.824	0.0789	0.0526
hard	55	11	0.200	0.364	0.1273	0.0182

The bound holds in every stratum, and its slack widens with difficulty: the realized harm is 1.00, 0.67 and 0.14 of the bound on easy, medium and hard problems. Both directions were registered before the comparator ran. The bound is tight exactly where the certificate is nearly sound, and conservative where it is not, because a request the certifier certifies wrongly on a hard problem is one the comparator is also likely to get wrong.

Table 18: The cross-family ladder: predicted against realized cost, over four comparator families and a 14B–70B scale range. Each entry is the realized cost ratio and, in parentheses, the signed miss against the ratio the realized-cost identity predicted *before any comparator token existed*, from the sealed 72B record alone. The authorization named 0.05 as the falsifier in advance: a larger miss narrows the predictive claim to within-family. Bold marks the one cell that exceeds it; daggers mark cells frozen by the efficiency gate. **Every cross-family cell in this table is a mechanism-role measurement. The cross-family authorizations do not permit a confirmation role to be opened, and no confirmation role was opened for any cell. No cell therefore adds a certified claim, whether its efficiency gate passed or failed.**

comparator	Multi-IF 5 lang.	Multi-IF Fr/Es	IFBench classic	IFBench OOD	JSON Schema GitHub	JSON Schema other
phi-4 (14.66B)	**0.453 (-0.134)**	0.420 (-0.041)	0.497 (-0.000)	0.868 (-0.009)†	0.555 (+0.006)	0.408 (+0.002)
Mistral-Small-24B (23.57B)	0.424 (-0.049)	0.348 (-0.005)	0.364 (-0.032)	0.733 (-0.037)†	0.442 (-0.002)	0.301 (-0.001)
Yi-1.5-34B (34.39B)	0.367 (-0.046)	0.299 (+0.002)	0.329 (-0.016)	0.702 (-0.012)†	0.364 (-0.025)	0.220 (-0.029)
Nemotron-70B (70.55B)	0.308 (-0.038)	0.230 (-0.005)	0.276 (-0.010)	0.634 (-0.018)	0.334 (+0.005)	0.187 (-0.002)

23 of 24 realized ratios fall inside the registered band (largest miss inside: 0.049); exactly one exceeds it. 21 gates passed and 3 froze, and the frozen cells are exactly the out-of-distribution IFBench subset, two of which were pre-registered as expected ceiling failures. Per-request cap violations are zero across all 24 cells. The identity's *scale* term transfers exactly, since α rescales with the comparator's parameter count; its *selection* term $\kappa\tau$, measured against one comparator's generation lengths, does not describe another family's, which is what the single miss records.

remaining budget, so that more calls to the large model are truncated. The fourth prediction fails too, and widely: a certifier from another family at essentially the same scale is accepted on barely half of the requests on which the 4B is accepted, a gap of nearly twenty-five points where the falsifier was ten. The benchmark subset is five-language instruction following, which the Qwen family handles well and that family does not, so the number establishes that a cross-family certifier is not interchangeable here, not that scale is all that matters anywhere.

Routers and cascades on the cells with a verifier. Table 9 in Appendix E runs eight routing and cascade alternatives under the per-request budget on the two official routing pools, where the policy calls no verifier, and scores a ninth, AvengersPro, from the decisions of its official implementation, with the same zero charge for every router's own forward pass. Table 21 runs the same eight alternatives on the 22 cross-family cells with a verifier, the cells of the ladder that remain after excluding two on which the large model is below 3% accurate. The comparison is not information-symmetric: the policy reads the benchmark's exact per-request verifier and no alternative is given it. The row that carries the attribution is therefore the uncapped control of Section 4, the same-information control of the table: the certificate with the prefix truncation and the middle stage removed. That control recovers essentially all of the cost and accuracy advantage over cascade-routing on its own and breaks the per-request budget on 212 requests; what the mechanism adds is the guarantee, since 212 violations become none.

Table 19: The admission condition tested out of sample. The corrected cost rule of Proposition 4(b) was derived after seeing which within-family cells failed and scored 11 of 11 there as a fit; the admission record registered the cross-family ladder as its test before that ladder ran. Each entry is the realized cost ratio of the three-stage policy (the sealed Qwen3-32B admitted as a middle certifier) against the two-stage ratio in parentheses, so a lower first number means the middle stage paid. The mark records the two rules' verdicts: ✓ both right, † classical index wrong and corrected rule right. Comparators are ordered by parameter count. An asterisk marks a cell on which the three-stage arm exceeds the per-request budget on at least one request; on such a request no external prefix fits, so the terminal contributes no recorded cost and the printed three-stage ratio understates what the policy would spend.

subset	phi-4 (14.7B)	Mistral-24B (23.6B)	Yi-1.5-34B (34.4B)	Nemotron-70B (70.6B)
Multi-IF 5 lang.	0.723 (0.453)$^{\checkmark *}$	0.598 (0.424)$^{\checkmark *}$	0.369 (0.367)$^{\checkmark *}$	**0.282 (0.308)**†
Multi-IF Fr/Es	0.607 (0.420)$^{\checkmark *}$	0.438 (0.348)$^{\checkmark *}$	0.316 (0.299)$^{\checkmark *}$	**0.202 (0.230)**$^{\checkmark}$
IFBench classic	0.829 (0.497)$^{\checkmark *}$	0.490 (0.364)$^{\checkmark *}$	0.389 (0.329)$^{\checkmark *}$	**0.264 (0.276)**†
IFBench OOD	1.630 (0.868)$^{\checkmark *}$	1.077 (0.733)$^{\checkmark *}$	0.984 (0.702)$^{\checkmark *}$	0.800 (0.634)$^{\checkmark}$
JSON Schema GitHub	0.928 (0.555)$^{\checkmark *}$	0.578 (0.442)$^{\checkmark *}$	**0.358 (0.364)**†	**0.288 (0.334)**$^{\checkmark}$
JSON Schema other	0.585 (0.408)$^{\checkmark *}$	0.362 (0.301)$^{\checkmark *}$	**0.208 (0.220)**$^{\checkmark}$	**0.148 (0.187)**$^{\checkmark}$

Out of sample the corrected rule is right on 24 of 24 cells and the classical index on 21; they disagree on 3, and the corrected rule is right on each. The middle stage pays on 7 cells, all against the two largest terminals, and never against phi-4 or Mistral-24B: exactly where the terminal is dear relative to the 32B. Three-stage accuracy improved on every cell, so the accuracy rule is not discriminating here, as the registration warned. The success curves the accuracy rule consumes were measured on the 72B terminal and aliased by subset; the cost rule uses the paired records' own costs. The asterisk marks the 16 of 24 cells whose three-stage arm breaks the per-request budget, on 872 requests in all.

Table 20: Moving the certifier, everything else fixed. One subset (Multi-IF turn 1, five languages), the same 726 mechanism-role requests, the same sealed Qwen3-32B comparator bank replayed for every arm; only the certifier checkpoint changes. α is the certifier's average parameter-weighted tokens over the comparator's, q_1 its certificate rate, and *ratio* the realized aggregate cost. *Predicted* carries $\kappa\tau$ unchanged from the 4B cell, as registered. **Mechanism role only: no gate, no confirmation role and no certified claim.**

certifier	parameters	α	q_1	ratio	predicted	miss
Qwen3-0.6B	596,049,920	0.0118	0.4573	0.5948	0.7022	0.1073
Qwen3-1.7B	1,720,574,976	0.0537	0.5730	0.5259	0.5968	0.0709
Qwen3-4B *	4,022,468,096	0.1401	0.7741	0.4274	0.4274	0.0000
Qwen3-14B	14,768,307,200	0.4374	0.7920	0.6468	0.7020	0.0552
Phi-4-mini-instruct ‡	3,836,021,760	0.1029	0.5262	0.5830	–	–

*the incumbent certifier of the eight confirmed cells; its bank is the sealed one, replayed, and $\kappa\tau$ is solved on it, so its miss is zero by construction. ‡a certifier from another family, outside the size ladder, so no predicted ratio is formed for it. Registered scorecard: **A1 holds** (no adjacent pair inverts; $+11.57$, $+20.11$, $+1.79$ points). **A3 holds** (not monotone; the minimum falls at Qwen3-4B, an interior point named in advance). **A2 falsified**: carrying $\kappa\tau = 1.2720$ from the 4B misses by up to 0.1073 against a registered falsifier of 0.05, on 3 of the four arms. The identity cannot fail; what fails is the assumption that $\kappa\tau$ *transfers*. Its implied value runs 1.0067 to 1.2720 and the 4B is the maximum, so the carried factor over-predicts every other arm. **A4 falsified**: the cross-family certifier (‡), at essentially the same scale, certifies on 52.62% of requests against the 4B's 77.41%, a gap of 24.79 points where the registered falsifier was ten. Two things belong beside that number. Its first run was void, generated with a stop set that omitted this model's own turn terminator, and would have put the gap at 40.50 points: the defect exaggerated it by about sixteen, so the re-run changed the magnitude and not the verdict. This subset is five-language instruction following, where the Qwen family is strong and this one is not, so what is measured is family at fixed scale *on a multilingual subset*; a single-language subset might narrow it. The stop-set correction was verified to be a no-op for Qwen on all 726 rows, so the arms remain like-for-like.

Four predictions were registered before any alternative ran, and two of them did not hold. The policy was predicted to be cheaper than every alternative that violates no cap. It is not. The certifier pass rate of a cell is the fraction of its mechanism-role requests on which the certifier's answer is accepted, and it exceeds one half on 18 of the 22 cells; on 14 of those 18 cells each of the two cost-aware routers has zero cap violations and a lower cost than the two-stage policy, and each is less accurate than the policy on average over the 22 cells. The record supports non-domination, and it supports neither dominance nor lowest cost: on none of the 22 cells is the large model or any of the eight alternatives at least as cheap and at least as accurate as the two-stage policy while strictly better

in cost or in accuracy. The second prediction was that some alternative would fall below the large model on at least 6 of the 22 cells while the policy fell below it on none. Its second half held and its first failed: the alternative that fell below the large model most often did so on 2 cells. On average over the 22 cells the large model is the least accurate arm of the table, tied only with majority voting and the FrugalGPT-style cascade, which reduce to it, so few alternatives had room to fall below it; that is a limitation of the cells and not a property of the alternatives. The third prediction holds for the two-stage policy and fails for the three-stage policy, an ambiguity in the registration rather than a result, and it is reported in all three readings rather than resolved after the fact. The fourth holds: cascade-routing matches or beats the two-stage policy in accuracy on 4 of the 22 cells, against an allowance of 11. The certifier is more accurate than the large model on 19 of the 22 cells; it is not on IFBench multi-turn constraints at turn 1 against Nemotron-70B, on JSONSchemaBench at turn 1 against Mistral-24B, and on JSONSchemaBench at turn 1 against phi-4. On all 22 cells the acceptance rate equals the certifier's own accuracy, because the verifier is the scoring metric.

AvengersPro on the cells with a verifier. Table 21 reports the eight alternatives that were registered before any of them ran, and AvengersPro is not one of them. It was added afterwards, under a registration sealed before the router had produced a decision on these cells and labelled post hoc, so it adds no certified claim and is reported here rather than in that table. The official selection runs unchanged: the published clustering router, its published constants and the same GTE-Qwen2-7B embeddings that the other embedding-fed alternatives read, with the router's forward pass charged zero, as for every other router. Before any of its numbers were read, the eight sealed alternatives were recomputed on the same inputs and reproduced the sealed per-group summaries field for field, and the router's decisions joined to the sealed requests with no disagreement between its own recorded outcome and the recorded outputs. Two predictions were registered. The first fails: AvengersPro breaks the per-request budget on 701 requests, spread over 12 of the 22 cells, and every one of those 12 cells has Mistral-24B or phi-4 as its large model, the two cheapest large models of the ladder, where the per-request budget is tightest; on the cap-2048 group of cells against phi-4 its cost ratio is 1.233 and on the cap-2048 group against Mistral-24B it is 1.018, so on those groups it spends more than the large model it is meant to spend less than. The second holds: on none of the 22 cells is AvengersPro at least as accurate as the policy at no greater cost with zero violations, under the two-stage reading or under the three-stage reading, and the hindsight reading, which selects one of those two policies on each cell, leaves no cell either.

Comparison arms on the confirmation roles. Every alternative in Tables 9, 21, 13 and 14 was run on a development role. On the confirmation roles of the eight confirmed Qwen3-32B cells, the sealed comparison is of the policy with the certifier alone and with the large model alone, and with no other arm. Table 22 adds, on those roles, every further arm whose inputs already exist: the single model selected on the mechanism role, random routing, random escalation, the uncapped control and the hindsight oracle. Each entry is computed exactly from the recorded per-request outcomes and costs of those roles. The replay runs no model, opens no role and evaluates no gate, so it cannot change a verdict; it is a held-out comparison on already-opened confirmation requests, descriptive, and adds no confirmed claim.

Against every deployable arm that does not read the certificate, the policy attains a higher accuracy on every confirmation role. The policy exceeds the accuracy of the large model by 0.75 to 8.32 points. It exceeds the accuracy of the certifier by 7.20 to 11.35 points, and it exceeds the accuracy of the single model selected on the mechanism role by 0.75 to 9.01 points. It exceeds the expected accuracy of random routing by 6.23 to 8.79 points, and it exceeds the expected accuracy of random escalation by 4.23 to 9.44 points. On Multi-IF five languages, turn 1, and on Multi-IF English, French and Spanish, turn 2, the model selected on the mechanism role is the certifier, while the more accurate single model in hindsight is the large model. On every other cell the model selected on the mechanism role is the more accurate single model in hindsight.

Against the uncapped control, which reads the same certificate, the policy attains no higher accuracy on any confirmation role. On IFBench classic constraints and on both JSONSchemaBench collections, the control and the policy certify the same requests. On each Multi-IF cell the control certifies 1 to 6 more requests than the policy. An exact two-sided sign test on the requests that only one of the policy and the control certifies gives $p < 0.05$ only on Multi-IF English, French and Spanish, turn 3. There the control alone certifies six requests, and $p = 0.031$ without correction for testing on every

Table 21: Routing and cascade baselines on the *certificate* cells, the counterpart to Table 9, which runs the same eight arms from the same implementation on the two routing pools where no exact verifier exists. Means over the 22 cross-family cells that remain after excluding two on which the comparator is below 3% accurate. Cost is the aggregate ratio to the protected comparator; violations are per-request budget breaches before the response-safe wrapper. The baselines see only a correctness predictor fitted on frozen embeddings; **none of them is given the exact certificate**, so the honest attribution needs the same-information control: the certificate with our two mechanisms removed. It recovers almost all of the accuracy and cost advantage on its own and *breaks the per-request budget on 212 requests*. What the mechanism buys here is the guarantee, not the cost.

policy	cost	accuracy	cap violations
Protected comparator	1.000	0.6500	0
Majority vote (Chen et al., 2024a) †	1.000	0.6500	0
FrugalGPT cascade (Chen et al., 2024a) †	1.000	0.6500	0
AutoMix self-verification (Aggarwal et al., 2024)	0.871	0.6806	485
Embedding router (Ong et al., 2025)	0.923	0.7795	974
Matrix-factorization router (Ong et al., 2025)	0.919	0.7779	968
Matrix-factorization, cost-aware	0.329	0.7553	0
Embedding router, cost-aware	0.333	0.7562	0
Cascade-routing (Dekoninck et al., 2025)	0.652	0.7788	0
Certifier alone, every request ‡	0.153	0.7445	0
Certificate only, no truncation, no middle stage	0.432	0.8140	212
Certificate-first policy (ours)	**0.433**	**0.8135**	**0**

†These tasks have no finite public answer alphabet, so the answer-agreement arms have no agreement to compute and reduce to the protected comparator on every cell. Our row is the two-stage policy on every cell, with no per-cell choice between the two-stage policy and the three-stage policy. The three-stage policy is reported separately because it breaks the per-request budget on 16 of the 24 cross-family cells. The certifier pass rate of a cell is the fraction of the cell's mechanism-role requests on which the certifier's output passes the certificate. The certifier pass rate exceeds one half on 18 of the 22 cells in the means. On 14 of the 18 cells with certifier pass rate above one half, the cost-aware embedding router and the cost-aware matrix-factorization router each have zero cap violations and a lower cost than our policy. Averaged over the same cells, the accuracy of our policy exceeds the accuracy of the cost-aware embedding router by 5.40 percentage points. Averaged over the same cells, the accuracy of our policy exceeds the accuracy of the cost-aware matrix-factorization router by 5.49 percentage points. On none of the 22 cells in the means is the protected comparator or any routing or cascade arm both at least as cheap and at least as accurate as our policy. ‡The certifier's own answer delivered on every request, under the same request budget and wrapper;

cell. The control exceeds the per-request budget on 331 requests, from 1 on JSONSchemaBench other collections to 124 on Multi-IF five languages, turn 2; the policy exceeds it on none. Without the response-safe rule, the cost ratio of the control exceeds that of the policy on every cell, by 0.0007 to 0.0286. The response-safe rule replaces the answer of the control on a violating request by the answer of the large model and charges the cost of the large model alone. With that rule, the cost ratio of the control is below that of the policy on every cell except JSONSchemaBench GitHub and JSONSchemaBench other collections. As in Table 21, the verifier without truncation attains at least the accuracy gain of the policy over the large model. Truncation under the remaining budget removes the budget violations of the control, lowers the cost ratio without the response-safe rule by at most 0.0286, and lowers the accuracy by at most 0.66 points. Each arm that would need new model output is listed below the table with the reason.

Learned routers on the confirmation roles. The seven arms of Table 22 that need a request embedding were then run on the same already-opened confirmation requests, under a registration sealed before any embedding existed: every router setting was fitted on each cell's fit role, the GTE-Qwen2-7B embedder of Table 9 produced the request embeddings after the seal, and the comparison was scored once. It is a held-out comparison on already-opened confirmation requests, descriptive, and adds no confirmed claim. Of the 56 pairs of arm and cell (Table 23), no arm is more accurate than the policy and no arm has a positive exact 95% lower bound over it (Table 24); the difference is positive on all 56 pairs and its lower bound is positive on 51, the exceptions being the five pairs of JSONSchemaBench other collections at $+0.75$ points. Seven pairs have an arm that is cheaper than

Table 22: Comparison arms replayed on the confirmation roles of the eight confirmed certificate cells (Qwen3-4B certifier, Qwen3-32B comparator). Every entry is exact arithmetic on the sealed per-request outcomes and costs of those roles: no model is run, no role is opened, and no entry can change a verdict. **Held-out comparison on already-opened confirmation requests; descriptive; adds no confirmed claim.** Accuracy is the percentage of requests that the cell's official checker accepts. Cost is parameter-weighted tokens divided by the comparator's on the same requests, so the comparator's cost is 1. *comp.*: the comparator. *cert.*: the certifier, delivered unconditionally; the larger of the comparator and certifier accuracies is the best single model in hindsight. *sel.*: the single model that certified more mechanism-role requests (C: certifier, P: comparator), delivered on every request, with that model's accuracy and cost. *rand. rout.*: random routing, which chooses the comparator with the probability that makes its expected cost equal to the policy's. *rand. esc.* and *oracle*: as in Table 13; the oracle's accuracy equals that of *no trunc.* by construction. *no trunc.*: the certificate-only control of Table 21, with its cost taken after the response-safe wrapper. v: the number of requests on which that control exceeds the per-request budget before the wrapper.

			accuracy (%)						cost					
cell	n	sel.	comp.	cert.	rand. rout.	rand. esc.	no trunc.	policy	cert.	rand. esc.	no trunc.	policy	oracle	v
Multi-IF 5 languages, turn 1	1,166	C	79.25	76.84	77.67	78.99	86.11	85.85	0.133	0.364	0.432	0.432	0.409	72
Multi-IF 5 languages, turn 2	1,166	C	58.58	58.75	58.66	62.25	67.24	66.90	0.125	0.538	0.576	0.577	0.545	124
Multi-IF Fr/Es, turn 1	495	P	87.07	83.84	84.44	85.27	92.73	92.53	0.136	0.298	0.297	0.297	0.279	7
Multi-IF En/Fr/Es, turn 2	933	C	68.49	67.74	68.04	70.30	75.67	75.35	0.128	0.450	0.481	0.482	0.448	43
Multi-IF En/Fr/Es, turn 3	912	P	55.59	51.21	53.86	55.86	60.75	60.09	0.125	0.613	0.651	0.654	0.607	78
IFBench classic	861	P	83.04	80.72	81.21	82.11	87.92	87.92	0.122	0.314	0.308	0.308	0.286	3
JSONSchemaBench GitHub	731	P	92.89	83.17	85.74	85.08	94.53	94.53	0.124	0.292	0.355	0.355	0.327	3
JSONSchemaBench other	534	P	96.82	88.95	90.30	89.90	97.57	97.57	0.124	0.234	0.274	0.274	0.255	1

the policy with zero budget violations: the six arms other than AutoMix on Multi-IF five languages, turn 2, at 8.49 to 8.83 points lower accuracy, and AvengersPro on Multi-IF English, French and Spanish, turn 3, at 7.57 points lower accuracy. With two candidate models the quality-first router, the cost-aware router and cascade-routing apply the same decision rule to the same predictor and therefore coincide on every cell; their columns are identical by construction. The AutoMix arm exceeds the per-request budget on 1 to 181 requests per cell. Multi-IF turns 2 and 3 feed the router the earlier turns' user messages and this turn's prompt, never the large model's own answers.

Serving verifiable and non-verifiable requests by different policies, on one stream. A registered development run asked whether one stream is served better by sending the requests that admit an exact verifier to certificate stopping and the remaining requests to a learned router, than by either policy alone. The registration was sealed before any output of this pool existed. The pool is Qwen3-1.7B, Qwen3-4B-Instruct-2507 and Qwen3-14B with Qwen3-32B as the large model. The verifiable part is the training and validation split of JSONSchemaBench, which no cell of this paper had opened, and the non-verifiable part is the seven LLMRouterBench contexts that no cell of this paper had opened. The certifier and every router setting were fitted on the fit role, the rule that serves the non-verifiable part was fitted there as well, and the router received the budget that the verifiable part did not spend. Scoring ran once, on the mechanism role and on the validation role.

The registered conjunction failed on both roles, so the confirmation role stayed closed and the run is a frozen negative. On the verifiable part the composite is more accurate than the router, 94.70 points against 93.17 on the mechanism role, and that comparison passes its registered test on both roles. On the non-verifiable part the composite is less accurate than certificate stopping alone and costs more, 61.70 points at 0.883 of the large model's cost against 67.44 at 0.539, and it is also less accurate than the same router at its own budget, which reaches 67.96 at 0.719. Against the large model the composite therefore reaches 76.66 points at 0.568 on the mechanism role, below the large model's 77.96, and it exceeds the per-request budget on 32 requests, so the efficiency gate fails on the accuracy bound and on the violation count together. The validation role repeats every verdict. Read against the reference that serves each part by the policy registered for it, the composite loses 2.44 points on the mechanism role and 3.84 on the validation role. The transfer of the unspent budget, which is what makes this design more than an arithmetic identity over the two parts, is also what fails: the router placed at the transferred budget is worse than the same router at its own.

Table 23: **Held-out comparison on already-opened confirmation requests; descriptive; adds no confirmed claim.** The seven embedding-fed routing arms of Table 21, fitted on each cell's fit role and scored on the already-opened confirmation requests of the eight confirmed certificate cells (Qwen3-4B certifier, Qwen3-32B comparator), beside the sealed policy on the same requests. Each entry is accuracy (percentage of requests the cell's official checker accepts) at cost ratio (parameter-weighted tokens divided by the comparator's on the same requests), with the number of requests whose decision exceeded the per-request budget in parentheses; over-budget decisions fall back to the comparator's answer at the comparator's cost, as in Table 9. Every router forward pass, including the 7B embedding call, is charged zero here; Table 25 charges it. The arms that need no embedding are in Table 22 and are not re-scored.

cell	n	policy	LR router, quality-first	LR router, cost-aware	MF router, quality-first	MF router, cost-aware	cascade-routing	AutoMix	AvengersPro
Multi-IF 5 lang., t1	1,166	85.85 @ 0.432 (0)	79.67 @ 0.774 (0)	79.67 @ 0.774 (0)	78.47 @ 0.782 (0)	78.47 @ 0.782 (0)	79.67 @ 0.774 (0)	79.42 @ 0.968 (134)	79.33 @ 0.553 (0)
Multi-IF 5 lang., t2	1,166	66.90 @ 0.577 (0)	58.32 @ 0.522 (0)	58.32 @ 0.522 (0)	58.40 @ 0.472 (0)	58.40 @ 0.472 (0)	58.32 @ 0.522 (0)	58.15 @ 1.025 (181)	58.06 @ 0.546 (0)
Multi-IF Fr/Es, t1	495	92.53 @ 0.297 (0)	87.27 @ 0.987 (0)	87.27 @ 0.987 (0)	87.27 @ 0.977 (0)	87.27 @ 0.977 (0)	87.27 @ 0.987 (0)	86.67 @ 1.015 (20)	84.85 @ 0.565 (0)
Multi-IF En/Fr/Es, t2	933	75.35 @ 0.482 (0)	68.92 @ 0.865 (0)	68.92 @ 0.865 (0)	68.17 @ 0.855 (0)	68.17 @ 0.855 (0)	68.92 @ 0.865 (0)	68.38 @ 0.988 (79)	68.49 @ 0.571 (0)
Multi-IF En/Fr/Es, t3	912	60.09 @ 0.654 (0)	53.84 @ 0.705 (0)	53.84 @ 0.705 (0)	53.73 @ 0.720 (0)	53.73 @ 0.720 (0)	53.84 @ 0.705 (0)	53.95 @ 1.045 (106)	52.52 @ 0.593 (0)
IFBench classic	861	87.92 @ 0.308 (0)	83.62 @ 0.928 (0)	83.62 @ 0.928 (0)	82.93 @ 0.943 (0)	82.93 @ 0.943 (0)	83.62 @ 0.928 (0)	83.86 @ 0.753 (9)	82.35 @ 0.785 (0)
JSONSchemaBench GitHub	731	94.53 @ 0.355 (0)	92.89 @ 0.995 (0)	92.89 @ 0.995 (0)	93.16 @ 0.992 (0)	93.16 @ 0.992 (0)	92.89 @ 0.995 (0)	92.75 @ 0.781 (4)	92.89 @ 0.910 (0)
JSONSchemaBench other	534	97.57 @ 0.274 (0)	96.82 @ 1.000 (0)	96.82 @ 1.000 (0)	96.82 @ 1.000 (0)	96.82 @ 1.000 (0)	96.82 @ 1.000 (0)	95.13 @ 0.560 (1)	94.94 @ 0.604 (0)

Of the 56 arm–cell pairs, 0 have an arm more accurate than the policy, 0 have a positive exact 95% lower bound on the arm minus the policy (Table 24), and 7 have an arm cheaper than the policy with zero budget violations.

Table 24: **Held-out comparison on already-opened confirmation requests; descriptive; adds no confirmed claim.** Policy minus arm on the confirmation requests of Table 23, in accuracy points, with the exact one-sided 95% lower bound on the paired difference in each direction: ℓ_{p} bounds policy minus arm from below and ℓ_{a} bounds arm minus policy from below, both by the lattice Clopper–Pearson bound that the confirmation records themselves use (support $\{-1, 0, 1\}$, $\alpha = 0.05$). A positive ℓ_{a} marks an arm that is more accurate than the policy on that cell with the same exact guarantee; a negative entry in the first column marks an arm that is more accurate in the sample.

cell	LR router, quality-first Δ (ℓ_{p}, ℓ_{a})	LR router, cost-aware Δ (ℓ_{p}, ℓ_{a})	MF router, quality-first Δ (ℓ_{p}, ℓ_{a})	MF router, cost-aware Δ (ℓ_{p}, ℓ_{a})	cascade-routing Δ (ℓ_{p}, ℓ_{a})	AutoMix Δ (ℓ_{p}, ℓ_{a})	AvengersPro Δ (ℓ_{p}, ℓ_{a})
Multi-IF 5 lang., t1	+6.17 (+4.19, -8.09)	+6.17 (+4.19, -8.09)	+7.38 (+5.26, -9.41)	+7.38 (+5.26, -9.41)	+6.17 (+4.19, -8.09)	+6.43 (+4.42, -8.37)	+6.52 (+4.71, -8.23)
Multi-IF 5 lang., t2	+8.58 (+6.35, -10.72)	+8.58 (+6.35, -10.72)	+8.49 (+6.27, -10.63)	+8.49 (+6.27, -10.63)	+8.58 (+6.35, -10.72)	+8.75 (+6.40, -11.01)	+8.83 (+6.64, -10.94)
Multi-IF Fr/Es, t1	+5.25 (+2.28, -8.05)	+5.25 (+2.28, -8.05)	+5.25 (+2.28, -8.05)	+5.25 (+2.28, -8.05)	+5.25 (+2.28, -8.05)	+5.86 (+3.00, -8.53)	+7.68 (+4.50, -10.63)
Multi-IF En/Fr/Es, t2	+6.43 (+4.11, -8.66)	+6.43 (+4.11, -8.66)	+7.18 (+4.77, -9.49)	+7.18 (+4.77, -9.49)	+6.43 (+4.11, -8.66)	+6.97 (+4.58, -9.26)	+6.86 (+4.77, -8.84)
Multi-IF En/Fr/Es, t3	+6.25 (+3.79, -8.63)	+6.25 (+3.79, -8.63)	+6.36 (+3.95, -8.69)	+6.36 (+3.95, -8.69)	+6.25 (+3.79, -8.63)	+6.14 (+3.63, -8.57)	+7.57 (+5.15, -9.87)
IFBench classic	+4.30 (+2.48, -6.02)	+4.30 (+2.48, -6.02)	+4.99 (+3.06, -6.82)	+4.99 (+3.06, -6.82)	+4.30 (+2.48, -6.02)	+4.07 (+2.28, -5.75)	+5.57 (+3.56, -7.48)
JSONSchemaBench GitHub	+1.64 (+0.24, -2.96)	+1.64 (+0.24, -2.96)	+1.37 (+0.06, -2.61)	+1.37 (+0.06, -2.61)	+1.64 (+0.24, -2.96)	+1.78 (+0.34, -3.14)	+1.64 (+0.24, -2.96)
JSONSchemaBench other	+0.75 (-0.58, -2.02)	+0.75 (-0.58, -2.02)	+0.75 (-0.58, -2.02)	+0.75 (-0.58, -2.02)	+0.75 (-0.58, -2.02)	+2.43 (+0.47, -4.28)	+2.62 (+0.60, -4.52)

Table 25: **Held-out comparison on already-opened confirmation requests; descriptive; adds no confirmed claim.** The arms of Table 23 with the embedding call charged, as in Table 10: 7 billion parameters times the input tokens the embedder reads, at most 7,500 per request, on every request. Each entry is accuracy at cost ratio, with the number of requests whose charged decision exceeded the budget in parentheses.

cell	policy	LR router, quality-first	LR router, cost-aware	MF router, quality-first	MF router, cost-aware	cascade-routing	AutoMix	AvengersPro
Multi-IF 5 lang., t1	85.85 @ 0.432 (0)	79.67 @ 0.806 (90)	79.67 @ 0.806 (90)	78.47 @ 0.814 (99)	78.47 @ 0.814 (99)	79.67 @ 0.806 (90)	79.42 @ 0.999 (137)	79.33 @ 0.585 (68)
Multi-IF 5 lang., t2	66.90 @ 0.577 (0)	58.32 @ 0.545 (71)	58.32 @ 0.545 (71)	58.40 @ 0.495 (66)	58.40 @ 0.495 (66)	58.32 @ 0.545 (71)	58.15 @ 1.046 (194)	58.06 @ 0.569 (74)
Multi-IF Fr/Es, t1	92.53 @ 0.297 (0)	87.27 @ 1.014 (16)	87.27 @ 1.014 (16)	87.27 @ 1.005 (15)	87.27 @ 1.005 (15)	87.27 @ 1.014 (16)	86.67 @ 1.042 (21)	84.85 @ 0.593 (10)
Multi-IF En/Fr/Es, t2	75.35 @ 0.482 (0)	68.92 @ 0.883 (48)	68.92 @ 0.883 (48)	68.17 @ 0.874 (51)	68.17 @ 0.874 (51)	68.92 @ 0.883 (48)	68.38 @ 1.006 (81)	68.49 @ 0.589 (30)
Multi-IF En/Fr/Es, t3	60.09 @ 0.654 (0)	53.84 @ 0.720 (48)	53.84 @ 0.720 (48)	53.73 @ 0.736 (46)	53.73 @ 0.736 (46)	53.84 @ 0.720 (48)	53.95 @ 1.059 (112)	52.52 @ 0.609 (37)
IFBench classic	87.92 @ 0.308 (0)	83.62 @ 1.007 (8)	83.62 @ 1.007 (8)	82.93 @ 1.022 (8)	82.93 @ 1.022 (8)	83.62 @ 1.007 (8)	83.86 @ 0.829 (16)	82.35 @ 0.864 (3)
JSONSchemaBench GitHub	94.53 @ 0.355 (0)	92.89 @ 1.087 (4)	92.89 @ 1.087 (4)	93.16 @ 1.084 (4)	93.16 @ 1.084 (4)	92.89 @ 1.087 (4)	92.75 @ 0.872 (5)	92.89 @ 1.002 (4)
JSONSchemaBench other	97.57 @ 0.274 (0)	96.82 @ 1.078 (1)	96.82 @ 1.078 (1)	96.82 @ 1.078 (1)	96.82 @ 1.078 (1)	96.82 @ 1.078 (1)	95.13 @ 0.638 (1)	94.94 @ 0.682 (0)

Three readings of the same run belong beside that negative. Certificate stopping alone reaches 79.80 points at 0.419 of the large model's cost with no budget violation, above the large model on both roles, which is the result this paper reports elsewhere and not a new claim here. Placing certificate stopping before the router, with admission on the complete cost cap and the large model replayed under the remaining budget, reaches 79.92 points at 0.533 with no budget violation, and that arm is descriptive. Every one of the composite's 32 budget violations on the mechanism role, and of its 30 on the validation role, is a request that fits the budget until the router's embedding call is charged, so the violation component of the gate is decided by that accounting rather than by a second model call. The layering test registered for this pool passes for four of the eight routers on each role. One audit identified the rows whose question overlaps a release that this project had already read, and a second audit flags 9 of the 6,466 rows whose question text is contained in a prompt that a pool model had already answered in this project. The rows flagged by either audit were kept in the registered

reading and removed in a separately scored reading, and no verdict of the three registered tests or of the layering test changes on either role under either charging rule.

Locators for Table 5. Each row states the problem as the paper's own text states it, read in full: FrugalGPT, Sections 2 and 3 (Chen et al., 2024a); cascade-routing, Equation (1), Definition 2 and Theorem 2 (Dekoninck et al., 2025); RouteLLM, Equations (1) and (2) (Ong et al., 2025); Hybrid LLM, Sections 2.2 and 3 (Ding et al., 2024); AutoMix, Sections 3 and 4.2 (Aggarwal et al., 2024); Avengers-Pro, Section 3 (Zhang et al., 2025b); C3PO, Equations (2) and (3) and Section 4.2 (Valkanas et al., 2025); ThriftLLM, Definition 2 and Sections 3.2 and 4.3.1 (Huang et al., 2025). Hybrid LLM has no arm in Tables 9 and 21; the RouteLLM-style routers are the closest arms. The AutoMix arm substitutes the fit-role correctness predictor for self-verification because the released records carry no self-verification score. The reported experiments do not provide matched comparisons with ThriftLLM or C3PO. ThriftLLM prunes calls when the remaining models cannot change its aggregate prediction; it does not require an exact correctness certificate. C3PO uses confidence estimated from repeated responses. The single-response records used here do not supply that input. Reproducing either method on its released data does not resolve this comparison limitation.

Sensitivity to the ceiling and to the prefix stride. The efficiency gate uses a ceiling of 0.70 on the cost ratio, preregistered rather than derived, and the replay of the large model's answer uses a prefix boundary stride of 16 tokens. Both are recomputed here from the recorded outcomes with every other gate condition at its sealed value, and both recomputations are post hoc descriptions that change no verdict. Table 26 recomputes every gate outcome over 21 values of the ceiling from 0.60 to 0.80. The number of confirmed rows that Table 6 would hold rises from 20 to 25 across that range, stands at 23 at the preregistered 0.70, and never falls as the ceiling rises. Eleven cells change outcome somewhere in $[0.65, 0.75]$ and are listed with their realized ratios in Table 27, and six rows of Table 6 carry gates that contain no ceiling. Table 28 replays the ten Qwen3-32B cells, which contain the eight confirmed cells, at strides of 16, 32, 64, 128 and 256 tokens. Eight cells are confirmed and two are frozen at every one of those strides, so no certificate in this paper turns on the registered stride. A stride that is not a multiple of 16 would need prefix texts that were never frozen and never scored, so its accuracy is not computed. Its cost is computable from token counts alone, and at strides of 1, 2, 4 and 8 tokens on the same ten cells the recomputed cost ratios change no ceiling clause at 0.70.

Repeatability of the instruction and schema checkers. This audit uses previously opened development requests only, and it changes no verdict and no gate. The JSONSchemaBench checkers were called on the delivered answers and on the stored prefixes of the large model, three times per check: the GitHub collection contributes 680 distinct requests and 1,360 request-action rows, the other collections contribute 533 distinct requests and 1,066 request-action rows, and every verdict matched its stored value. IFBench contributes 840 distinct development requests and 1,680 request-action rows. Three repetitions of 3,737 checks produced 11,211 calls, of which 15 disagreed with a stored strict or loose verdict, affecting four requests. There were no execution errors. Identical inputs also changed verdict within the repetitions. A diagnostic restricted to those affected requests then fixed the language-detector seed at each of 0, 1 and 2. For each seed, ten repetitions of twelve checks were stable; four checks changed across seeds. This is a post hoc diagnostic of the checker, not a new accuracy estimate. The original outcomes and the exclusion rules are unchanged. For a randomized checker, equality of the verifier and the metric programs alone does not imply that a fresh metric call accepts every answer that the verifier previously accepted. The exactness condition of Definition 1 is a pointwise implication for fixed maps. It applies to reused recorded verdicts or to a fixed shared checker state that satisfies the implication, not automatically to independent fresh seeds. The recorded cost identities remain identities, but a fresh randomized re-evaluation requires its own error analysis.

G ARCHITECTURE-AWARE FLOP LEDGER

Section 2 meters the cost of an open checkpoint in parameter-weighted tokens (PWT), the product of the model's parameter count and the number of tokens it processes. This appendix gives the analytic FLOP estimate that audits that cost on the measured paths. The estimate is an audit; it is not an admission rule of any policy.

Table 26: Gate outcomes as a function of the aggregate cost ceiling θ, recomputed from the sealed mechanism and confirmation records with every other gate condition at its sealed value. This is a post hoc description: every certificate in the paper uses $\theta = 0.70$, the row in bold. *Mech.* counts the cells whose mechanism gate passes; *conf.* counts the cells whose confirmation role passes the efficiency gate, and a cell that passes only the improvement gate is not counted, as in the efficiency ledger; *unop.* counts the cells whose mechanism gate passes but whose confirmation role was never opened, so that no confirmation outcome exists for them. The Qwen2.5-72B columns cover the ten subsets of Table 8; the Goedel-Prover-V2-32B column covers the MiniF2F and ProofNet cells; the cross-family column covers the mechanism-role cells of Table 18. The last column counts the confirmed rows the efficiency ledger would hold; it includes the rows for HEADLINES, SimpleQA, MBPP, the 128-task MiniF2F cell, IFEval-es and M-IFEval, whose own gates contain no ceiling.

	Qwen3-32B			Goedel-32B	Qwen2.5-72B, two stages			Qwen2.5-72B, three stages			cross-family	ledger
θ	mech.	conf.	unop.	conf.	mech.	conf.	unop.	mech.	conf.	unop.	mech.	confirmed
0.60	7	7	0	0	8	7	0	8	7	0	20	20
0.61	8	7	0	0	8	7	0	8	7	0	20	20
0.62	8	7	0	0	8	7	0	8	7	0	20	20
0.63	8	7	0	0	8	7	0	8	7	0	20	20
0.64	8	7	0	0	8	8	0	8	7	0	20	21
0.65	8	7	0	0	8	8	0	8	7	0	20	21
0.66	8	7	0	0	8	8	0	8	8	0	21	21
0.67	8	7	0	0	10	9	0	8	8	0	21	22
0.68	8	8	0	0	10	9	0	8	8	0	21	23
0.69	8	8	0	0	10	9	0	8	8	0	21	23
0.70	8	8	0	0	10	9	0	8	8	0	21	23
0.71	8	8	0	1	10	9	0	9	8	1	21	24
0.72	9	8	1	1	10	9	0	9	8	1	22	24
0.73	10	8	2	1	10	9	0	9	8	1	22	24
0.74	10	8	2	1	10	10	0	9	8	1	22	25
0.75	10	8	2	1	10	10	0	9	8	1	23	25
0.76	10	8	2	1	10	10	0	9	8	1	23	25
0.77	10	8	2	1	10	10	0	9	8	1	23	25
0.78	10	8	2	1	10	10	0	9	8	1	23	25
0.79	10	8	2	1	10	10	0	9	8	1	23	25
0.80	10	8	2	1	10	10	0	9	8	1	23	25

The FLOP ledger is a deterministic analytic estimate, not a measurement from a hardware profiler. It counts one multiply-add as two FLOPs, charges the LM head for every processed token, and uses full causal attention geometry. The PWT and FLOP ledgers cover model inference only. They exclude verifier, parsing, prompt construction, sandbox, scheduling, and orchestration work, which must be profiled separately when end-to-end latency is claimed. The full-attention geometry and the every-token LM-head convention deliberately upper-count common cached inference. The estimate depends only on the input length I and the output length O: for the models considered, the operation count of a forward pass is set by tensor shapes, not by token values, and a fixed number of experts per token keeps this true for a mixture-of-experts model. The ledger defines the comparison cost; it does not claim to reproduce kernel-specific wall-clock operations. The parametric term follows the compute scaling of (Kaplan et al., 2020); the attention terms use the transformer's dot-product geometry (Vaswani et al., 2017).

For input length $I \geq 1$ and output length $O \geq 1$, define

$$N(I,O) := I + O, \qquad G(I,O) := \frac{I(I+1)}{2} + IO + \frac{O(O+1)}{2}. \tag{4}$$

Here N is the number of processed tokens and G is the number of causal context pairs under full attention. Let $h, V, L, H_q, H_{kv}, d, m$ denote hidden width, vocabulary size, layer count, query-head count, KV-head count, head dimension, and intermediate width. Put $d_q = H_q d$ and $d_{kv} = H_{kv} d$.

Proposition 14 (Qwen2/3 FLOP estimate)**.** *For Qwen2 or Qwen3, let*

$$A_Q = h(d_q + 2d_{kv}) + d_q h, \qquad M_Q = 3hm. \tag{5}$$

The FLOP count is

$$\begin{aligned} \mathfrak{F}_{\mathrm{Qwen}}(I,O) = {} & 2L(A_Q + M_Q)N(I,O) + 4Ld_qG(I,O) \\ & + L\{16h + 8m + 6(d_q + d_{kv})\}N(I,O) \\ & + 5LH_qG(I,O) + 2hVN(I,O). \end{aligned} \tag{6}$$

Table 27: Every cell whose gate outcome at some $\theta \in [0.65, 0.75]$ differs from its outcome at $\theta = 0.70$, from the record of Table 26. Each ratio is the realized aggregate cost ratio on the role, followed by the ratios on its two halves; the last column gives each outcome that differs from the outcome at 0.70 and the values of θ in $[0.65, 0.75]$ at which it holds.

cell	comparator (policy)	mechanism ratio (halves)	confirmation ratio (halves)	outcome at 0.70	other outcomes in $[0.65, 0.75]$
Multi-IF turn 3, five languages	Qwen3-32B (two stages)	0.676 (0.639/0.715)	–	frozen at mechanism gate	mechanism gate passes, never opened at $\theta \in [0.72, 0.75]$
Multi-IF turn 3, English, French and Spanish	Qwen3-32B (two stages)	0.553 (0.507/0.602)	0.654 (0.671/0.635)	confirmed	improvement gate only at $\theta \in [0.65, 0.67]$
IFBench, out-of-distribution constraints	Qwen3-32B (two stages)	0.712 (0.729/0.693)	–	frozen at mechanism gate	mechanism gate passes, never opened at $\theta \in [0.73, 0.75]$
MiniF2F (valid mechanism role, test confirmation role)	Goedel-Prover-V2-32B (Lean)	0.533 (0.495/0.568)	0.692 (0.679/0.704)	no certificate	confirmed at $\theta \in [0.71, 0.75]$
Multi-IF turn 3, five languages	Qwen2.5-72B (two stages)	0.628 (0.590/0.669)	0.715 (0.736/0.692)	no certificate	frozen at mechanism gate at $\theta \in [0.65, 0.66]$; confirmed at $\theta \in [0.74, 0.75]$
IFBench, out-of-distribution constraints	Qwen2.5-72B (two stages)	0.651 (0.667/0.632)	0.633 (0.614/0.651)	confirmed	frozen at mechanism gate at $\theta \in [0.65, 0.66]$
Multi-IF turn 3, five languages	Qwen2.5-72B (three stages)	0.669 (0.630/0.710)	–	frozen at mechanism gate	mechanism gate passes, never opened at $\theta \in [0.71, 0.75]$
Multi-IF turn 3, English, French and Spanish	Qwen2.5-72B (three stages)	0.509 (0.441/0.579)	0.655 (0.654/0.656)	confirmed	improvement gate only at $\theta = 0.65$
IFBench, out-of-distribution constraints	Mistral-Small-24B (two stages, mechanism role)	0.733 (0.749/0.716)	–	gate fails	gate passes at $\theta = 0.75$
IFBench, out-of-distribution constraints	Nemotron-70B (two stages, mechanism role)	0.634 (0.653/0.614)	–	gate passes	gate fails at $\theta = 0.65$
IFBench, out-of-distribution constraints	Yi-1.5-34B (two stages, mechanism role)	0.702 (0.720/0.684)	–	gate fails	gate passes at $\theta \in [0.72, 0.75]$

The five terms are block linear maps, attention matrix products, tokenwise elementwise work, pairwise softmax work, and the LM-head allowance. The shared elementwise expression is the Qwen3 allowance and therefore also covers the smaller Qwen2 Q/K-normalization workload.

Proof. The Q, K, V, and output projections contain $h(d_q + 2d_{kv}) + d_q h = A_Q$ parameters, and the three SwiGLU linear maps contain $3hm = M_Q$. Multiplying these maps by every processed token and counting a multiply-add twice gives the first term. The two attention matrix products give $4Ld_qG$. The RMSNorm, Q/K normalization, RoPE, SwiGLU, and softmax allowances give the next two terms. Applying the vocabulary head to all N processed tokens gives $2hVN$. Summing the five terms gives Equation (6). □

Proposition 15 (LFM2 FLOP estimate)**.** *Let $\widetilde{m}$ be the LFM2 intermediate width, that is, the feed-forward width after the optional $2/3$ adjustment, the optional multiplier, and upward rounding to a multiple of a fixed block size. Let L_A be the number of full-attention layers and let $L_C = L - L_A$*

Table 28: Prefix boundary stride, replayed from the sealed Qwen3-32B score files. This is a post hoc description: every certificate uses the registered stride of 16 tokens, and the replay at stride 16 reproduces every sealed row. A stride that is a multiple of 16 keeps the boundaries at its multiples, every boundary whose frozen text ends with a newline, and the terminal. Each entry is the aggregate cost ratio on the role that decides the cell (the confirmation role where it was opened, the mechanism role otherwise), followed by the accuracy difference in points and the truncation factor τ of Proposition 1; a dagger marks an entry whose gate outcome at $\theta = 0.70$ differs from the outcome at stride 16. Strides that are not multiples of 16 need prefix texts that were never frozen and never scored, and are not computed.

cell	role	stride 16	stride 32	stride 64	stride 128	stride 256
Multi-IF turn 1, five languages	confirmation	0.432 (+6.60, 0.941)	0.431 (+6.60, 0.939)	0.431 (+6.60, 0.938)	0.427 (+6.60, 0.925)	0.426 (+6.60, 0.924)
Multi-IF turn 2, five languages	confirmation	0.577 (+8.32, 0.940)	0.576 (+8.32, 0.939)	0.575 (+8.32, 0.936)	0.574 (+8.32, 0.935)	0.570 (+8.23, 0.926)
Multi-IF turn 3, five languages	mechanism	0.676 (+4.29, 0.949)	0.676 (+4.29, 0.948)	0.675 (+4.29, 0.947)	0.673 (+4.29, 0.944)	0.672 (+4.29, 0.942)
Multi-IF turn 1, French and Spanish	confirmation	0.297 (+5.45, 0.972)	0.297 (+5.45, 0.970)	0.297 (+5.45, 0.969)	0.296 (+5.45, 0.965)	0.296 (+5.45, 0.965)
Multi-IF turn 2, English, French and Spanish	confirmation	0.482 (+6.86, 0.965)	0.481 (+6.86, 0.964)	0.481 (+6.86, 0.962)	0.480 (+6.86, 0.961)	0.479 (+6.86, 0.959)
Multi-IF turn 3, English, French and Spanish	confirmation	0.654 (+4.50, 0.959)	0.653 (+4.50, 0.958)	0.653 (+4.50, 0.957)	0.652 (+4.50, 0.956)	0.651 (+4.50, 0.954)
IFBench, classic constraints	confirmation	0.308 (+4.88, 0.995)	0.308 (+4.88, 0.995)	0.308 (+4.88, 0.995)	0.308 (+4.88, 0.995)	0.308 (+4.88, 0.995)
IFBench, out-of-distribution constraints	mechanism	0.712 (+5.13, 0.993)	0.712 (+5.13, 0.993)	0.712 (+5.13, 0.993)	0.712 (+5.13, 0.993)	0.712 (+5.13, 0.993)
JSONSchemaBench, GitHub collections	confirmation	0.355 (+1.64, 0.994)	0.355 (+1.64, 0.994)	0.355 (+1.64, 0.994)	0.355 (+1.64, 0.994)	0.355 (+1.64, 0.994)
JSONSchemaBench, other collections	confirmation	0.274 (+0.75, 0.995)	0.274 (+0.75, 0.995)	0.274 (+0.75, 0.995)	0.274 (+0.75, 0.994)	0.274 (+0.75, 0.994)

be the number of convolution layers. Let κ be the kernel length of the short convolution in each convolution layer, that is, the number of past positions that the convolution keeps. Define

$$A_L = h(d_q + 2d_{kv}) + d_q h, \qquad M_L = 3h\widetilde{m}, \qquad Q_L = 4h^2. \tag{7}$$

Then

$$\begin{aligned}\mathfrak{F}_{\mathrm{LFM2}}(I,O) = 2\{LM_L + L_A A_L + L_C Q_L\}N(I,O) + 4L_A d_q G(I,O) \\ + L(12h + 8\widetilde{m})N(I,O) + 6L_A(d_q + d_{kv})N(I,O) \\ + 5L_A H_q G(I,O) + L_C(2h\kappa + 4h)N(I,O) \\ + 2hVN(I,O).\end{aligned} \tag{8}$$

Only the full-attention layers incur the quadratic attention terms.

Proof. Every layer contributes three MLP maps. A full-attention layer contributes the Q/K/V/O projection count A_L, whereas each convolution layer contributes the four $h \times h$ convolution projections Q_L. The second term counts attention matrix products only for the L_A full-attention layers. The next three lines are the norm, activation, positional, softmax, convolution, and LM-head allowances. Summing the components gives Equation (8). □

H HARDNESS OF CHOOSING THE CERTIFIER SET

Section 2 proposes no rule for choosing which certifiers to run. This appendix states the complexity of that choice in its simplest form: every certifier costs one unit, acceptance is deterministic, and at most k certifiers may be chosen. This form is a special case of choosing the certifier set under a shared budget.

Theorem 16 (Choosing the certifier set contains maximum coverage)**.** *Let $\mathcal{X}$ be a finite set of requests, and let the request X be drawn uniformly from $\mathcal{X}$. Let $a_1, \ldots, a_m$ be certifiers, where certifier a_j accepts exactly the requests in a set $S_j \subseteq \mathcal{X}$ and costs one unit. For a set $A \subseteq \{a_1, \ldots, a_m\}$ of at most k certifiers, let*

$$V(A) = \Pr\Big\{X \in \bigcup_{a_j \in A} S_j\Big\} = \frac{1}{|\mathcal{X}|}\Big|\bigcup_{a_j \in A} S_j\Big| \tag{9}$$

be the probability that at least one chosen certifier accepts the request. Choosing A to maximize $V(A)$ is NP-hard. Unless $\mathrm{P} = \mathrm{NP}$, for every fixed $\epsilon > 0$ no polynomial-time algorithm achieves approximation ratio $1 - 1/e + \epsilon$ uniformly over these instances.

Proof. Reduce from maximum-k-coverage. Given a universe U, sets $S_1, \ldots, S_m \subseteq U$, and a budget k, create one equally likely request x_e per element $e \in U$ and one unit-cost certifier a_j per set. The

verifier accepts certifier a_j on request x_e exactly when $e \in S_j$. For every chosen set A of at most k certifiers,

$$V(A) = \frac{1}{|U|} \left| \bigcup_{a_j \in A} S_j \right|. \tag{10}$$

Thus an exact optimizer solves maximum-k-coverage, and any better approximation transfers to that problem. The hardness and threshold follow from the maximum-coverage result of Feige (1998). □

The theorem concerns the choice of the set. It does not say that an exhaustive search over the orders of a small finite library is intractable, and it does not turn an ordered policy that stops at the first acceptance into a coverage problem.

I Details moved from Sections 3 and 4

Full limitations statement. Of the 21 routing contexts only Performance-20's MBPP has a public verifier, and the role split does not control pretraining exposure (Appendix A.6). Where the verifier is the official metric, the results certify cost under it, not quality that the checker does not measure. The 0.70 ceiling is preregistered, not derived; Appendix I varies it and the prefix stride. Metered cost is not latency or memory: at batch one, sealed logs give a median of 0.38 to $0.52\times$ the comparator's and a 99th percentile near $1.25\times$. We report nothing under concurrency because we did not measure it (Appendix I). Finite cells imply no population guarantee: each confirmed cell is one draw, its lower bound the uncertainty, and the nine 72B re-confirmations reuse their 32B prompts. Results depend on the registered pool, comparator and budget; the 32B and 72B certifiers are from the comparator's own family, and a larger model is not automatically a better large model. IFBench's multi-turn release, reproduced as published, leaves 73 rows with no deterministic verdict, and two checkers call an unseeded language detector. JSONSchemaBench's checker rejects all instances of three formats. Affected rows are excluded by rules sealed before confirmation. Repeated IFBench checks changed some verdicts (Appendix F); recorded scores do not cover fresh randomized runs. Multi-IF confirmation prompts translate prompts opened in English, so a cluster bootstrap accompanies the exact bound. A control from the certifier's family, selected like the policy, is more accurate than the policy on two confirmation roles: by 1.01 points on Multi-IF (interval $[-2.02, -0.02]$) at $1.06\times$ its tokens and by 0.78 points on IFBench ($[-1.36, -0.31]$) at $0.97\times$. They rest on released records and prices, certify those models on those dates and support no universal model-quality or scaling claim.

Model families. The models of Section 4 are from the Qwen3 (Yang et al., 2025), Qwen2.5 (Yang et al., 2024), Goedel-Prover-V2 (Lin et al., 2026), phi-4 (Abdin et al., 2024), Mistral Small (Mistral AI, 2025), Yi (Young et al., 2024) and Nemotron (Wang et al., 2025) families.

Cells sealed before the efficiency gate. Six cells sealed before the efficiency gate existed (IFEval-es, M-IFEval, the 128-task MiniF2F cell, HEADLINES, SNLI, the MBPP root) carry their own earlier gates, and three of them (HEADLINES, M-IFEval, SNLI) sit above $0.70\times$. HEADLINES ($+15.22$ at $0.799\times$) uses the registered external comparator, and SNLI ($+2.76$ at $0.904\times$) passed a composition-superiority gate, not the efficiency gate.

Vocabulary of the generated tables. Generated tables use earlier names: StateCompose for the policy of Algorithm 1, StateCompose root for its root rule, residual cap for the remaining budget, certificate rate for the acceptance rate, and protected comparator for the large model of Section 2; the text says comparator where several large models are compared in turn. The tables also use "certificate" for cell-level gates, whereas the text reserves it for the verdict. Statuses in Table 6: "mechanism gate failed" (failed on development requests), "independent role" (separate development pool), "opened transfer replay" (replay on opened requests) and "its own composition-superiority gate" (SNLI's earlier accuracy gate); the appendix's "failed closed" means that the large model was kept.

Certifier ablation. A certifier ablation on the Multi-IF turn-1 cell narrows it to the same certifier: the acceptance rate rises with certifier size at every step from 0.6B to 14B, the realized cost ratio

is lowest at the 4B, and carrying $\kappa\tau$ from the 4B to the other sizes over-predicts by up to 0.1073 against a registered falsifier of 0.05 (Table 20).

Concurrency. We report nothing under concurrency because we did not measure it: no available serving stack supported both the certifier's architecture and our cluster's driver, so the sweep was abandoned rather than approximated.

Sensitivity to the ceiling and the prefix stride. Recomputed over ceilings from 0.60 to 0.80, the number of confirmed rows runs from 20 to 25 and is 23 at 0.70, and no certificate turns on the prefix stride (Appendix F).

Knob-family sweep. On cross-family development cells, placing certificate stopping before every knob family tested raised accuracy by 4.42 to 7.47 points at a matched budget, with the router's embedding call charged; part of that gain holds by construction because the verifier is the scoring metric.

Routing-pool details. On the API pool the arms other than majority voting and random routing lie between 71.23 and 74.75 in macro accuracy, and they separate on cost. Cascade-routing runs at $0.587\times$ the large model's cost, below our root policy's $0.682\times$; it is below the large model in macro accuracy and above it per request. AvengersPro reaches 74.20 macro at $0.901\times$ with one cap violation, above our fit-selected root's 73.07, and 68.27 macro at $1.203\times$ with none, 1.20 points above the large model; both cost ratios exceed the 0.70 ceiling, and charging the embedding call raises them (Table 10). On SimpleQA the FrugalGPT cascade and majority voting read the released grade as an answer; with the large model kept on SimpleQA their exact lower bounds are -1.11 and -3.86 points rather than $+5.20$ and $+2.74$ (Appendix E). Replayed on the held-out confirmation requests of the eight confirmed cells, which adds no confirmed claim, that control exceeds the budget on 331 requests and the policy on none. The 22 development cells contain $10{,}040$ pooled requests: 4×523 from IFBench multi-turn classic, 4×409 from IFBench out-of-distribution, 4×726 from Multi-IF five languages, 4×309 from Multi-IF French and Spanish, and 3×415 and 3×309 from the two JSONSchemaBench collections.

HEADLINES: the best single member. On HEADLINES the best single member trails the policy by 6.80 points.