Knowing When to Ask for Help: Bayesian Self-Escalation in Hierarchical LLM Agents*

Nadeem Shaikh
Independent Researcher
Melbourne, Australia
nadeem@nadeemshaikh.net

August 26, 2026

Abstract

Current LLM agent systems decide delegation *before* reasoning begins (a router picks a model) or *after* a response is complete (a verifier scores it and may retry). We study a third regime: an agent that recognises, *during* its own reasoning, that it is unlikely to succeed and transfers control to a stronger model. Our contribution is not the observation that agents can defer to other agents—that is well established—but a decision-theoretic formulation of *intra-generation* delegation as an optimal-stopping problem over an online estimate of the agent's eventual task success. The junior agent maintains a *competence posterior*: an abstract Bayesian state whose sufficient statistics are *learned* from labelled trajectories, not read off raw entropy. It escalates when the expected cost of continuing exceeds the expected cost of deferral. We derive the myopic escalation threshold in closed form, characterise the optimal policy via dynamic programming, and prove—using monotonicity alone, without any concavity argument—that under a conditional-independence signal model the optimal policy is a time-varying threshold on the competence posterior, with no monotone-likelihood or other shape assumption on the raw signal. We further prove that the oracle belief separates at the Chernoff-information rate of the signal, and give a finite-sample guarantee: with n labelled calibration trajectories of length T, the deployed plug-in policy's regret over the oracle is $O\big(Lt\sqrt{\log K/(nT)}\big)$ with high probability. Our central message is a regret bound showing that excess cost is controlled by the *calibration* of that posterior: better calibration matters more than a smarter router, and "confidently wrong" predictions are the binding failure mode. A simulation study with a known data-generating process confirms each prediction of the theory. A pre-registered protocol yields falsifiable predictions; in this revised version we report a first execution of that protocol on a `Qwen2.5-Coder` 1.5B→7B code cascade, confirming two of its three predictions: the escalation frontier dominates post-hoc routing at equal cost, and the cumulative competence belief's discrimination rises over the course of generation. The theoretical results are unchanged.

Keywords: LLM agents; hierarchical inference; model cascades; optimal stopping; uncertainty quantification; calibration; selective prediction; learning to defer.

*__Version 1.1.__ Version 1 (DOI: 10.5281/zenodo.21330788) reported no real-model results and pre-registered an evaluation protocol in their place. This version adds a first execution of that protocol on a `Qwen2.5-Coder` 1.5B→7B code cascade (Section 8), confirming two of its three predictions; the theoretical results are unchanged, and the constant-threshold instantiation tested does not exercise the optimal-stopping dynamic program.

1 Introduction

The economics of large language model (LLM) deployment increasingly favour *hierarchies*: a small, fast model handles the bulk of traffic while a larger, more capable—and considerably more expensive—model is invoked only when needed. Model cascades and routers realise this idea and can cut inference cost by large factors without sacrificing quality [1, 10, 19]. Almost all such systems, however, make the routing decision either *before* inference (a router inspects the query and picks a model) or *after* a full generation (a verifier scores the completed output and may retry). Neither option lets the working model notice, partway through a difficult chain of reasoning, that it has left the region of competence and should hand off.

Many systems already let an agent hand work to another agent or to itself: self-reflection and critic/debate models, verifier-guided generation and best-of-n reranking, speculative decoding with fallback, and adaptive-compute and test-time-scaling methods all revisit or reallocate computation. What is missing across these is a *principled, sequential* account of *when*, mid-generation, to stop and defer. We do not claim to introduce agent delegation. We introduce a **decision-theoretic formulation of intra-generation delegation in which escalation is an optimal-stopping problem over an online estimate of eventual task success**. We refer to the mechanism as **self-escalation**.

A point we make central rather than incidental: the framework does *not* assume that raw uncertainty metrics are calibrated. High entropy is not failure and low entropy is not correctness—a model can be confidently wrong. The Bayesian state is therefore an *abstract competence posterior* whose sufficient statistics (the mapping from a signal history to a probability of success) are *learned from labelled trajectories*. Per-token entropy or the next-token margin serve only as raw evidence into that learned map, never as the posterior itself. This distinction is what makes the theory meaningful, and it turns out to be the crux: our main result is that the method's cost is governed by how well that posterior is calibrated.

Contributions.

1. **Formulation.** We cast intra-generation self-escalation as a Bayesian optimal-stopping problem over the junior agent's eventual success (Section 3), separating a *learned* competence-posterior component from the decision component.

2. **Policy and theory.** We derive the myopic escalation threshold in closed form, characterise the optimal policy via dynamic programming, and prove (i) that the competence posterior is a martingale, (ii) that the optimal policy is a time-varying threshold—proved *via monotonicity alone, with no concavity argument and no monotone-likelihood assumption on the raw signal*—(iii) a regret bound linking excess cost to posterior miscalibration, (iv) exponential separation of the oracle belief at the Chernoff-information rate of the signal, and (v) a finite-sample regret guarantee for the plug-in policy, decaying as $1/\sqrt{n}$ in the number of labelled calibration trajectories (Sections 4–5).

3. **Calibration as the binding constraint.** We elevate the regret bound to a design principle: for self-escalation, improving the calibration of the competence posterior dominates improving the decision rule (Section 5.4).

4. **Algorithm.** We give a streaming inference algorithm and an offline backward-induction procedure for the threshold schedule, with a text-level (not hidden-state) context handoff (Section 6).

5. **Simulation, protocol, and real-model validation.** On a model with a *known* data-generating process we confirm each prediction of the theory (Section 7); we pre-register a falsifiable protocol (Section 9); and we execute a first instance of it on a `Qwen2.5-Coder` code cascade, confirming two of its predictions (Section 8).

2 Related Work

Selective prediction and learning to defer. The option to abstain rather than predict has a long history as *selective prediction* or classification with a reject option [2, 5, 6]. *Learning to defer* generalises abstention by routing the rejected input to an external expert and learning the predictor and the deferral rule jointly [17, 18, 24]. Our setting is an instance of deferral in which the "expert" is a stronger model, but with two differences that existing formulations do not address: the deferral decision is made *sequentially within a single generation* rather than once per input, and the object of the posterior is the junior model's own eventual success rather than a label.

Cascades and routing for LLMs. FrugalGPT introduced learned LLM cascades that query models in increasing order of cost and stop when a scorer judges the answer adequate [1]. RouteLLM learns a router from preference data [19]; agreement- and consistency-based cascades escalate when cheap models disagree [10, 23]; and Jitkrittum et al. [8] analyse when confidence-based deferral in a cascade is and is not sufficient. These methods act at the granularity of a whole response and typically require at least one completed cheap generation (and often several samples) before deciding. Self-escalation instead consumes the token-level signal as it streams and can stop mid-generation, trading a small amount of bookkeeping for the ability to abort a doomed trajectory early.

Uncertainty quantification in LLMs. A large literature estimates LLM uncertainty from output distributions. Semantic entropy clusters sampled generations by meaning and measures entropy over clusters, giving a strong hallucination signal [3, 13]; semantic-entropy probes approximate this from a single forward pass's hidden states [11]. Most of this work targets *post-hoc* detection on a completed generation. We use such signals as the *evidence stream* of a sequential decision, and our framework is agnostic to which signal is used: token entropy, next-token margin, a probe output, or semantic entropy can all serve as the per-step observation e_t.

Self-revision and adaptive computation. A separate line of work reallocates or revisits computation within a single system. Adaptive computation time lets a network learn how many internal steps to spend per input [7]; early-exit decoding such as CALM stops a token's computation when intermediate confidence suffices [21]; speculative decoding drafts with a cheap model and verifies with an expensive one, falling back on disagreement [14]; self-refinement iterates on a model's own output with self-feedback [16]; and test-time-scaling studies how to allocate extra inference compute optimally [22]. These share our motivation—spend more effort only when needed—but they revise, rerank, or extend *the same* model's computation rather than framing a sequential decision to hand off to a *stronger* model, and they generally lack an explicit stopping rule tied to a calibrated estimate of eventual task success. Relatedly, Kadavath et al. [9] show LLMs' self-evaluations of correctness carry real signal but are imperfectly calibrated—precisely the regime where Proposition 3 says the gains of self-escalation are won or lost.

Sequential testing. The oracle decision core of our problem is a finite-horizon, cost-asymmetric relative of Wald's sequential probability ratio test, which continues sampling until the likelihood ratio exits an interval [25] and is optimal for the symmetric testing problem [26]. We flag this plainly so the reader can locate the classical core: the stopping mathematics is not new. What the classical theory does not supply—and where this paper works—is the formulation (deferral to a stronger model as the stopping action, with compute and error costs), the analysis when the likelihood ratio is *estimated* (Sections 5.4–5.5), and the systems instantiation.

Optimal stopping. The decision component is an optimal-stopping problem [4, 20]: at each step, stop-and-defer or continue-and-observe. We use standard monotone-stopping arguments to establish the threshold structure of the optimal policy. Our framing—an optimal-stopping rule over a *learned* online estimate of the working model's own eventual correctness, used to trigger escalation to a stronger model—combines these ingredients in a way we have not seen made explicit for LLM hierarchies, though we make no claim to the underlying stopping theory itself.

3 Problem Formulation

Consider a junior agent J and a senior agent S. Given a query x, agent J generates a response over T steps (tokens or reasoning steps). Let $Y \in \{0,1\}$ be the latent event that J's *completed* answer would be correct; write $\pi = \mathbb{P}(Y = 1)$ for the base competence of J on the query population. At each step $t \in \{1, \dots, T\}$ the agent emits a *competence-evidence* signal $e_t \in \mathcal{E}$ (a raw statistic such as normalised token entropy or the next-token margin). Let $\mathcal{F}_t = \sigma(e_1, \dots, e_t)$ and define the *competence posterior*

$$B_t \;=\; \mathbb{P}(Y = 1 \mid \mathcal{F}_t). \tag{1}$$

Remark 1 (The posterior is learned, not read off entropy). We stress that e_t is *evidence*, not the posterior. A raw uncertainty metric is not itself a likelihood ratio: high entropy is not failure and low entropy is not correctness, since a model can be confidently wrong. The mapping from a signal history to B_t—the sufficient statistics below—is *learned from labelled trajectories* and post-hoc calibrated (Section 6). Nothing in the theory requires the raw signal to be calibrated; it requires the fitted posterior to be. Proposition 3 makes the cost of failing this precise.

Assumption 1 (Conditional independence)**.** *Conditional on $Y = y$, the signals $(e_t)_{t \le T}$ are i.i.d. with density f_y, and f_0, f_1 have common support with finite log-likelihood ratio $\lambda(e) = \log \frac{f_1(e)}{f_0(e)}$. In practice λ is the learned statistic of Remark 1, not a closed-form entropy transform.*

Assumption 1 is a modelling idealisation; real token signals are correlated and non-stationary. We adopt it to obtain a transparent update and discuss its relaxation in Section 10. Under Assumption 1 the belief evolves as a simple log-odds recursion.

Lemma 1 (Belief update)**.** *Let $\ell_t = \log \frac{B_t}{1-B_t}$ and $\ell_0 = \log \frac{\pi}{1-\pi}$. Then $\ell_t = \ell_0 + \sum_{s=1}^{t} \lambda(e_s)$ and $B_t = \sigma(\ell_t)$ with $\sigma(z) = (1 + e^{-z})^{-1}$.*

Proof. By Bayes' rule and conditional independence, $\frac{\mathbb{P}(Y=1|\mathcal{F}_t)}{\mathbb{P}(Y=0|\mathcal{F}_t)} = \frac{\pi}{1-\pi} \prod_{s=1}^{t} \frac{f_1(e_s)}{f_0(e_s)}$; taking logs gives the recursion, and inverting the log-odds gives $B_t = \sigma(\ell_t)$. □

Costs and actions. Let $\kappa > 0$ be the compute cost per junior step, $\gamma > 0$ the one-off cost of escalation (senior compute plus added latency), and $L > 0$ the cost of a wrong final answer. The

senior returns a correct answer with probability $q \in (0,1)$, assumed (for the base model) independent of $\mathcal{F}_t$. At each step the agent chooses an action $a_t \in \{\text{CONTINUE}, \text{ESCALATE}\}$. Escalation is terminal and hands S the query together with a distilled context; if the agent never escalates it answers locally at step T. A policy ρ maps $\mathcal{F}_t$ to actions; we seek ρ minimising the expected total cost

$$\mathcal{C}(\rho) = \mathbb{E}\left[\underbrace{\kappa\, \tau_{\text{stop}}}_{\text{junior compute}} + \underbrace{\gamma\, \mathbf{1}\{\text{escalate}\}}_{\text{deferral}} + \underbrace{L\, \mathbf{1}\{\text{final answer wrong}\}}_{\text{error}} \right], \tag{2}$$

where τ_{stop} is the number of junior steps taken before stopping.

4 The Escalation Policy

4.1 Myopic threshold

Consider the decision at step t with belief $b = B_t$, comparing immediate escalation against *finishing locally*. The forward cost of escalating is $\gamma + (1-q)L$ (sunk junior steps are ignored). The forward cost of continuing to completion is $(T-t)\kappa + (1-b)L$. Escalation is preferred iff

$$\gamma + (1-q)L \;<\; (T-t)\kappa + (1-b)L. \tag{3}$$

Rearranging isolates a belief threshold.

Proposition 1 (Myopic escalation threshold)**.** *Under the "finish locally" comparison, escalate at step t iff $B_t < \tau_t^{\text{myo}}$, where*

$$\tau_t^{\text{myo}} \;=\; q - \frac{\gamma}{L} + \frac{(T-t)\kappa}{L}. \tag{4}$$

In particular $\tau_T^{\text{myo}} = q - \gamma/L$, and τ_t^{myo} decreases in t.

Equation (4) is interpretable: one escalates more readily when the senior is reliable (large q), when errors are costly (large L), and when escalation is cheap (small γ). The term $(T-t)\kappa/L$ says that with many tokens still to pay for, local completion looks relatively expensive, nudging the threshold up.

4.2 Optimal stopping

The myopic rule ignores the *option value* of continuing: another token yields fresh evidence that may resolve the uncertainty without paying for escalation. The optimal policy is the solution of the dynamic program with value function $V_t(b)$ equal to the minimal expected cost-to-go at step t with belief b:

$$V_T(b) = \min\big\{\gamma + (1-q)L,\ (1-b)L\big\}, \tag{5}$$

$$V_t(b) = \min\Big\{ \underbrace{\gamma + (1-q)L}_{\text{escalate}},\ \underbrace{\kappa + \mathbb{E}\big[V_{t+1}(B_{t+1}) \,\big|\, B_t = b\big]}_{\text{continue}} \Big\}, \quad t < T, \tag{6}$$

where the transition is the Bayes update of Lemma 1 driven by the posterior-predictive signal $e_{t+1} \sim b\, f_1 + (1-b) f_0$. The optimal policy escalates at t iff the escalate branch attains the minimum in (6).

5 Theoretical Analysis

5.1 The belief is a martingale

Theorem 1 (Martingale property)**.** *Under Assumption 1, $(B_t, \mathcal{F}_t)_{t\le T}$ is a martingale: $\mathbb{E}[B_{t+1} \mid \mathcal{F}_t] = B_t$.*

Proof. $B_t = \mathbb{P}(Y=1 \mid \mathcal{F}_t) = \mathbb{E}[\mathbf{1}\{Y=1\} \mid \mathcal{F}_t]$. By the tower property, $\mathbb{E}[B_{t+1} \mid \mathcal{F}_t] = \mathbb{E}[\mathbb{E}[\mathbf{1}\{Y=1\} \mid \mathcal{F}_{t+1}] \mid \mathcal{F}_t] = \mathbb{E}[\mathbf{1}\{Y=1\} \mid \mathcal{F}_t] = B_t$. □

Theorem 1 is often misread. It does *not* say waiting is worthless. Because the belief is a martingale, waiting does not increase the *expected* belief—but it can increase expected *decision quality* by revealing information, and the value of that information comes precisely from the nonlinearity of the future value $\mathbb{E}[V(B_{t+1})]$. A stock price is a martingale yet options on it have value; likewise here, the option to continue has value even though $\mathbb{E}[B_{t+1} \mid \mathcal{F}_t] = B_t$. What the martingale property does give us is a clean structural handle for the monotone-stopping argument below.

5.2 Consistency and separation rate

How quickly does the oracle belief become decisive? The answer is governed by a single scalar property of the signal pair: its Chernoff information.

Proposition 2 (Exponential belief separation)**.** *Assume a finite signal alphabet and let*

$$C \;=\; -\log \min_{s\in[0,1]} \sum_k f_0(k)^s f_1(k)^{1-s} \tag{7}$$

be the Chernoff information of (f_0, f_1), with $C > 0$ iff $f_0 \neq f_1$. Then for every fixed $\tau \in (0,1)$ there is a constant $A_\tau < \infty$, independent of t, such that

$$\mathbb{P}\big(B_t \le \tau \mid Y=1\big) \;\le\; A_\tau\, e^{-tC}, \qquad \mathbb{P}\big(B_t \ge \tau \mid Y=0\big) \;\le\; A_\tau\, e^{-tC}.$$

Consequently $B_t \to \mathbf{1}\{Y=1\}$ almost surely, and the oracle's step-t thresholded decision errs with probability at most $A_\tau e^{-tC}$.

Proof. $B_t \le \tau$ iff $\sum_{s\le t} \lambda(e_s) \le c$ with $c = \log \frac{\tau}{1-\tau} - \log \frac{\pi}{1-\pi}$. For any $s > 0$, the Chernoff bound gives $\mathbb{P}\big(\sum \lambda \le c \mid Y=1\big) \le e^{sc}\big(\mathbb{E}_{f_1}[e^{-s\lambda}]\big)^t = e^{sc}\big(\sum_k f_1(k)^{1-s} f_0(k)^s\big)^t$. Minimising the base over $s \in [0,1]$ yields e^{-C} per step, with $A_\tau = e^{s^\star c}$ at the minimiser $s^\star$. The $Y=0$ side is symmetric with $\mathbb{E}_{f_0}[e^{s\lambda}] = \sum_k f_0(k)^{1-s} f_1(k)^s$, whose minimum over $s \in [0,1]$ is the same e^{-C}. Almost-sure convergence follows from Borel–Cantelli. □

The practical reading: the number of tokens the oracle needs before its escalation decision is reliable at level α is $t \gtrsim \log(A_\tau/\alpha)/C$. "How informative is this uncertainty signal?" is thus answered by one number, computable from the fitted class-conditionals, and comparable across candidate signals (entropy vs. margin vs. probe output) before any policy is built.

5.3 Optimality of a threshold policy

The key structural fact is that a larger current belief makes the *next* belief stochastically larger. This is the step a careful reader will demand a proof of, so we give one. Notably, *no monotone-likelihood-ratio or other shape assumption on the raw signal is needed*: the Bayes update depends on e only through its log-likelihood ratio $\lambda(e)$, and the law of the likelihood ratio under f_1 stochastically dominates its law under f_0 for *any* pair of densities. Related belief-monotonicity results appear in the partially observed MDP literature [12, 15].

Lemma 2 (FOSD-monotone belief transition)**.** *Under Assumption 1 alone, the map* $b \mapsto \mathrm{Law}(B_{t+1} \mid B_t = b)$ *is non-decreasing in the first-order stochastic dominance (FOSD) order: for* $b \le b'$ *and every non-decreasing* $h : [0,1] \to \mathbb{R}$, $\mathbb{E}[h(B_{t+1}) \mid B_t = b] \le \mathbb{E}[h(B_{t+1}) \mid B_t = b']$.

Proof. Write the update as $B_{t+1} = \sigma\big(\log \frac{b}{1-b} + \Lambda\big)$ where $\Lambda = \lambda(e)$ and the incoming signal has posterior-predictive density $m_b = bf_1 + (1-b)f_0$. Three steps.

(i) *The law of* Λ *under* f_1 *FOSD-dominates its law under* f_0. Let $R = f_1(e)/f_0(e)$, so $\Lambda = \log R$ and it suffices to prove the claim for R. For any $c \ge 1$: $\mathbb{P}_{f_1}(R \ge c) = \mathbb{E}_{f_0}[R\,\mathbf{1}\{R \ge c\}] \ge c\,\mathbb{P}_{f_0}(R \ge c) \ge \mathbb{P}_{f_0}(R \ge c)$. For any $c < 1$: $\mathbb{P}_{f_1}(R < c) = \mathbb{E}_{f_0}[R\,\mathbf{1}\{R < c\}] \le c\,\mathbb{P}_{f_0}(R < c) \le \mathbb{P}_{f_0}(R < c)$, hence again $\mathbb{P}_{f_1}(R \ge c) \ge \mathbb{P}_{f_0}(R \ge c)$. No assumption on f_0, f_1 beyond common support was used.

(ii) *The law of* Λ *under* m_b *is FOSD-non-decreasing in* b. For non-decreasing u, $\mathbb{E}_{m_b}[u(\Lambda)] = b\,\mathbb{E}_{f_1}[u(\Lambda)] + (1-b)\,\mathbb{E}_{f_0}[u(\Lambda)]$ is affine in b with slope $\mathbb{E}_{f_1}[u] - \mathbb{E}_{f_0}[u] \ge 0$ by (i).

(iii) *Chaining.* B_{t+1} is non-decreasing in b for fixed Λ and non-decreasing in Λ for fixed b. For non-decreasing h, the map $\Lambda \mapsto h\big(\sigma(\log \frac{b}{1-b} + \Lambda)\big)$ is non-decreasing, so

$$\mathbb{E}_{m_{b'}}\big[h(B_{t+1}(b', \Lambda))\big] \;\ge\; \mathbb{E}_{m_{b'}}\big[h(B_{t+1}(b, \Lambda))\big] \;\ge\; \mathbb{E}_{m_b}\big[h(B_{t+1}(b, \Lambda))\big],$$

the first inequality by pointwise monotonicity in b, the second by (ii). □

Remark 2 (Signal orientation). For interpretability one typically wants λ monotone in e (lower entropy $\Rightarrow$ evidence for success), and our simulations use such signals; but the theory does not require it, since only the induced law of $\lambda(e)$ enters the update.

Theorem 2 (Threshold structure)**.** *Under Assumption 1, for each* t *there exists a threshold* $\tau_t^\star \in [0,1]$ *such that the optimal policy escalates at step* t *iff* $B_t \le \tau_t^\star$. *At the horizon,* $\tau_T^\star = q - \gamma/L$.

Proof. The argument uses *monotonicity only*; we never invoke concavity, so no concavity-preservation step is required. We show by backward induction that each V_t is non-increasing in b.

Base case. $V_T(b) = (1-b)L$ is non-increasing.

Inductive step. Suppose V_{t+1} is non-increasing. The escalate value $\gamma + (1-q)L$ is a constant, hence non-increasing. For the continue value $C_t(b) = \kappa + \mathbb{E}[V_{t+1}(B_{t+1}) \mid B_t = b]$, Lemma 2 gives that $b \mapsto \mathrm{Law}(B_{t+1} \mid b)$ is non-decreasing in the FOSD order. Since V_{t+1} is non-increasing, its expectation against an FOSD-larger law is smaller, so C_t is non-increasing in b. As the pointwise minimum of two non-increasing functions, $V_t = \min\{\gamma + (1-q)L, C_t\}$ is non-increasing, closing the induction.

Threshold structure. Escalation is optimal at b iff $\gamma + (1-q)L \le C_t(b)$. Because C_t is non-increasing, this set is a lower interval $[0, \tau_t^\star]$ with $\tau_t^\star = \sup\{b : \gamma + (1-q)L \le C_t(b)\}$ (and $\emptyset$ read as $\tau_t^\star = 0$), which is exactly a threshold rule.

Terminal threshold. At $t = T$ there is no continuation: the choice is escalate at cost $\gamma + (1-q)L$ or answer at expected cost $(1-b)L$ (Eq. (5)). Escalation is optimal iff $\gamma + (1-q)L \le (1-b)L$, i.e. $b \le q - \gamma/L$, matching the myopic value of Proposition 1. □

Remark 3 (Why monotonicity, not concavity). An earlier route argues V_t is concave and reads off the threshold from concavity. That route is defensible—the pointwise minimum of concave functions is in fact concave—but the concavity-*preservation* step under the Bayesian update is delicate and invites attack. The monotonicity proof above needs strictly less (only Lemma 2) and yields the same threshold conclusion, so we prefer it.

Remark 4 (Thresholds need not be monotone in t). An earlier draft claimed $\tau_1^\star \le \cdots \le \tau_T^\star$. That claim is *false in general*, and we retract it. Counterexample: take $\kappa > \gamma + (1-q)L$, so a single further token costs more than full escalation. Then continuing is never optimal before the horizon and $\tau_t^\star =$

1 for all $t < T$, while $\tau_T^\star = q - \gamma/L < 1$: thresholds *decrease*. The direction of the schedule reflects a tug-of-war between two forces—remaining-token costs (which favour escalating early, pushing early thresholds *up*) and the option value of information (which favours continuing early, pushing early thresholds *down*)—and which force wins depends on (κ, γ, L, q) and the informativeness of the signal. In the small-κ regime of our simulation the option value dominates and the computed interior schedule is low and gently increasing (Section 7); in that regime the myopic rule, whose threshold (4) is largest early, over-escalates at the start of generation (Table 1 quantifies this). Nor is non-monotonicity confined to the $\kappa > \gamma + (1-q)L$ boundary: re-running the backward induction of Section 7 with $\kappa = 0.02$ (all else unchanged) yields a schedule of ≈ 1.0 for $t \le 20$, dipping to ≈ 0.20 near $t = 37$ and rising again to ≈ 0.30 by $t = 39$—grossly non-monotone in the interior.

5.4 Calibration is the binding constraint

We regard the following as the paper's central practical message. The decision rule of Theorem 2 is optimal *given* the posterior, but in deployment the posterior is estimated, inducing beliefs $\hat{B}_t$ that may differ from the true B_t. The next bound shows the excess cost is controlled entirely by that gap—so, for self-escalation, effort spent improving the *calibration* of the competence posterior dominates effort spent on a more elaborate router or decision rule.

Proposition 3 (Miscalibration regret)**.** *Fix step t and a threshold τ. Let the realised decision (continue to completion vs. escalate) use $\hat{B}_t$ and the oracle decision use the true B_t, both thresholding at τ. Let $g_t(b) = (T-t)\kappa + (1-b)L - \gamma - (1-q)L$ denote the forward-cost difference (*CONTINUE *minus* ESCALATE*) at true belief b, and let $D = \{B_t, \hat{B}_t \text{ on opposite sides of } \tau\}$ be the disagreement event. Then*

$$\mathrm{Regret}_t \;\le\; L\,\mathbb{E}\big[\,|B_t - \hat{B}_t|\,\big] \;+\; |g_t(\tau)|\,\mathbb{P}(D). \tag{8}$$

In particular, at the myopic threshold $\tau = \tau_t^{\mathrm{myo}}$ of Proposition 1, where g_t vanishes,

$$\mathrm{Regret}_t \;\le\; L\,\mathbb{E}\big[\,|B_t - \hat{B}_t|\,\big]. \tag{9}$$

Proof. Excess cost is incurred only on D, where it equals $|g_t(B_t)|$. The function g_t is affine with slope $-L$, so $|g_t(B_t)| \le |g_t(B_t) - g_t(\tau)| + |g_t(\tau)| = L|B_t - \tau| + |g_t(\tau)|$. On D the threshold τ lies between B_t and $\hat{B}_t$, hence $|B_t - \tau| \le |B_t - \hat{B}_t|$. Taking expectations over D and bounding $\mathbb{E}[\mathbf{1}_D|B_t - \hat{B}_t|] \le \mathbb{E}|B_t - \hat{B}_t|$ gives (8); $g_t(\tau_t^{\mathrm{myo}}) = 0$ gives (9). □

Remark 5 (The threshold restriction is essential). An earlier draft asserted the bound (9) for *arbitrary* τ; that claim is false, and we retract it. Counterexample (at the horizon, with the costs of Section 7): take $\tau = 0.5$, $B_t = 0.49$, $\hat{B}_t = 0.51$. The decisions disagree and the realised regret is $g_T(0.49) = 0.51 - 0.25 = 0.26$, while $L|B_t - \hat{B}_t| = 0.02$. The extra $|g_t(\tau)|\,\mathbb{P}(D)$ term in (8) is exactly the price of operating at a threshold where the two actions are not cost-indifferent; it vanishes at τ_t^{myo} and is small near it.

Proposition 3 formalises the "confidently wrong" failure mode: a query with $Y = 0$ whose signals mimic success drives $\hat{B}_t$ high while the true B_t is low, producing an $O(L)$ regret event precisely when it is most costly. Two consequences are worth stating plainly. First, *confidently wrong predictions are the central failure mode of self-escalation*: no threshold policy on a miscalibrated signal can recover the lost escalations. Second, the bound is a directive for practitioners—measure and minimise $\mathbb{E}|B_t - \hat{B}_t|$ (via reliability diagrams and post-hoc calibration) before tuning thresholds, because thresholds cannot compensate for a miscalibrated posterior. Section 7 measures this dependence directly.

The bound as stated involves the unobservable oracle belief B_t. It can be connected to *measurable* calibration quantities in both directions.

Corollary 1 (Excess Brier score controls regret)**.** *Let* $\mathrm{BS}(Z) = \mathbb{E}[(Y - Z)^2]$ *denote the Brier score of a* $[0,1]$*-valued,* $\mathcal{F}_t$*-measurable predictor* Z*. At the myopic threshold* $\tau = \tau_t^{\mathrm{myo}}$*,*

$$\mathrm{Regret}_t \;\le\; L\sqrt{\mathrm{BS}(\hat{B}_t) - \mathrm{BS}(B_t)}. \tag{10}$$

Proof. Since $B_t = \mathbb{E}[Y \mid \mathcal{F}_t]$ and $\hat{B}_t$ is $\mathcal{F}_t$-measurable, the cross term vanishes in $\mathbb{E}[(Y - \hat{B}_t)^2] = \mathbb{E}[(Y-B_t)^2]+\mathbb{E}[(B_t-\hat{B}_t)^2]$ (the calibration–refinement decomposition), so $\mathbb{E}[(B_t-\hat{B}_t)^2] = \mathrm{BS}(\hat{B}_t)-\mathrm{BS}(B_t)$. Combine with Proposition 3 via $\mathbb{E}|B_t - \hat{B}_t| \le \sqrt{\mathbb{E}[(B_t - \hat{B}_t)^2]}$ (Jensen). □

Corollary 1 turns the abstract bound into a training objective: $\mathrm{BS}(B_t)$ is a fixed property of the signal, so *minimising the Brier score of the fitted posterior directly minimises the regret bound.* This is why the protocol in Section 9 reports Brier score as a primary metric rather than a diagnostic afterthought.

Remark 6 (ECE is necessary but not sufficient). The (L_1) expected calibration error satisfies $\mathrm{ECE}(\hat{B}_t) = \mathbb{E}\big|\mathbb{E}[Y \mid \hat{B}_t] - \hat{B}_t\big| = \mathbb{E}\big|\mathbb{E}[B_t - \hat{B}_t \mid \hat{B}_t]\big| \le \mathbb{E}|B_t - \hat{B}_t|$ by the tower property and Jensen. So ECE *lower-bounds* the quantity that drives regret: a large ECE certifies a problem, but a small ECE does not certify safety, because $\hat{B}_t$ can be perfectly calibrated on average while ignoring information in $\mathcal{F}_t$ (poor *refinement*). Brier score, which penalises both calibration and refinement, is the right target; ECE is the right alarm.

5.5 Sample complexity of calibrated self-escalation

The bounds above take the fitted posterior as given. We now close the loop: how much labelled calibration data buys how much regret? The following gives a complete, finite-sample answer for the plug-in estimator on a discretised signal (discretisation is standard in implementations; K is the number of bins).

Theorem 3 (Finite-sample regret of the plug-in policy)**.** *Assume a finite signal alphabet of size* K *with* $f_y(k) \ge \varepsilon$ *for all* k, y*, and a labelled calibration set containing* m_y *token observations of class* y*; let* $m = \min(m_0, m_1)$ *and suppose* $m \ge (2/\varepsilon^2)\log(4K/\delta)$*. Let* $\hat{\lambda}$ *be the plug-in log-likelihood ratio of the empirical bin frequencies, with the prior* π *known, and let* $\hat{B}_t$ *be the resulting beliefs. Then with probability at least* $1 - \delta$ *over the calibration set, simultaneously for all* $t \le T$*,*

$$\mathbb{E}\big[\,|\hat{B}_t - B_t|\,\big] \;\le\; \frac{t}{\varepsilon}\sqrt{\frac{\log(4K/\delta)}{2m}}, \qquad \textit{and hence, at the myopic threshold,} \qquad \mathrm{Regret}_t \;\le\; \frac{L\,t}{\varepsilon}\sqrt{\frac{\log(4K/\delta)}{2m}}. \tag{11}$$

Proof. Hoeffding's inequality gives $\mathbb{P}\big(|\hat{f}_y(k) - f_y(k)| \ge u\big) \le 2e^{-2m_y u^2}$ per bin and class; a union bound over the $2K$ pairs with $u = \sqrt{\log(4K/\delta)/(2m)}$ leaves failure probability at most δ. On the success event, the hypothesis on m gives $u \le \varepsilon/2$, so $\hat{f}_y(k) \ge \varepsilon/2$ and, since $x \mapsto \log x$ is $(2/\varepsilon)$-Lipschitz on $[\varepsilon/2, \infty)$, $|\log \hat{f}_y(k) - \log f_y(k)| \le 2u/\varepsilon$ for each class, hence $\|\hat{\lambda} - \lambda\|_\infty \le 4u/\varepsilon$. The log-odds error after t updates is at most $t \cdot 4u/\varepsilon$ (π known), and σ is $\frac{1}{4}$-Lipschitz, so $|\hat{B}_t - B_t| \le tu/\varepsilon$ pointwise on the success event, hence also in expectation over trajectories. Combining with Proposition 3 at the myopic threshold gives the regret bound. □

Three remarks, in decreasing order of comfort. First, the rate: regret decays as $O\big(Lt\sqrt{\log K/(nT)}\,/\varepsilon\big)$ when the calibration set consists of n trajectories of length T (so $m \approx nT\min(\pi, 1-\pi)$); every trajectory contributes T token observations, which is why modest labelled sets suffice in practice. Second, the linear-in-t compounding is a worst case of the plug-in construction; recalibrating $\hat{B}_t$

Algorithm 1 Bayesian Self-Escalation (inference time)

Require: query x; junior J; senior S; thresholds $\{\tau_t\}$; ratio λ; prior log-odds ℓ_0

$\ell \leftarrow \ell_0$
for $t = 1$ to T **do**
  $(\text{token}_t, e_t) \leftarrow J.\textsc{step}(x)$ ▷ e_t: token entropy / margin / probe output
  $\ell \leftarrow \ell + \lambda(e_t)$; $B \leftarrow \sigma(\ell)$ ▷ $O(1)$ belief update
  if $B < \tau_t$ **then**
    $c \leftarrow \textsc{DistillContext}(\text{trace}_{1:t},\ \text{tool\_state},\ \text{evidence})$
    return $S.\textsc{solve}(x, c)$ ▷ escalate; hand off *text-level* artifacts, not hidden state
  end if
end for
return $J.\textsc{finalize}()$ ▷ answer locally

directly at each t (e.g. isotonic regression per step) targets $\mathbb{E}|\hat{B}_t - B_t|$ without the compounding and is what we recommend in deployment. Third, the $1/\varepsilon$ dependence is pessimistic: it charges for accuracy on low-mass bins that belief trajectories near the threshold rarely visit; a margin-weighted refinement is left to future work. Section 7 verifies the theorem's driver empirically: the belief error decays at the predicted $1/\sqrt{n}$ rate, while the realised cost gap sits far below the bound.

6 Algorithm

Algorithm 1 is the streaming inference procedure: a single junior forward pass, an $O(1)$ belief update per token, and an early exit when the belief crosses the schedule. Algorithm 2 computes the threshold schedule $\{\tau_t^\star\}$ once, offline, by backward induction on a discretised belief grid, using the calibrated likelihood ratio to Monte-Carlo the transition.

Context handoff is text, not activations. The junior and senior are in general different models—different architectures, tokenizers, and KV-cache layouts—so the senior cannot ingest the junior's hidden state. DistillContext therefore produces *text-level* artifacts: the partial reasoning trace generated so far, any scratchpad or intermediate results, the tool-call history and their returns, and retrieved evidence. This is a prompt the senior can consume directly, and it keeps the handoff model-agnostic. It also means the escalation cost γ should include the tokens re-read by the senior, which the protocol in Section 9 measures.

Fitting the competence posterior. The likelihood ratio λ (or, equivalently, a direct map from signal history to B_t) is fit offline on a labelled development set: run J on queries with known correctness Y, collect signal trajectories $e_{1:T}$, and either (i) estimate the class-conditional densities f_0, f_1 parametrically or by kernel methods and take their log-ratio, or (ii) fit a logistic model mapping cumulative signal features (running mean entropy, margin trend, spike counts) to $\mathbb{P}(Y = 1)$, which sidesteps density estimation. Either way the resulting beliefs are then calibrated post hoc (isotonic regression or temperature scaling), targeting the Brier score per Corollary 1. Reliability diagrams of $\hat{B}_t$ against empirical success at several t are the basic sanity check.

Why a posterior plus thresholds, not a directly learned policy? One could instead train a classifier that maps signals straight to escalate/continue. We prefer the factored design for three reasons. First, *modularity under changing costs*: the fitted posterior depends only on the model

Algorithm 2 Offline threshold schedule (backward induction)

Require: calibrated λ; costs (κ, γ, L, q); grid $\{b_j\}_{j=1}^{m}$; samples $\{e^{(k)}\}$

$V(b_j) \leftarrow \min\{\gamma + (1-q)L,\ (1-b_j)L\}$ for all j ▷ terminal value, Eq. (5)

for $t = T-1$ down to 1 **do**

  for each grid point b_j **do**

    propagate $\ell_j = \log \frac{b_j}{1-b_j}$ by $\lambda(e^{(k)})$ under both classes; form next beliefs $b'^{(k)}$

    $C(b_j) \leftarrow \kappa + b_j \overline{V(b'^{(k)}_{Y=1})} + (1-b_j)\overline{V(b'^{(k)}_{Y=0})}$

    $V(b_j) \leftarrow \min\{\gamma + (1-q)L,\ C(b_j)\}$

  end for

  $\tau_t^\star \leftarrow \max\{b_j : \gamma + (1-q)L \le C(b_j)\}$

end for

return $\{\tau_t^\star\}$

and signal, while (κ, γ, L, q) enter only through Algorithm 2; when prices, latency budgets, or the senior model change, one reruns a cheap backward induction instead of recollecting labels and retraining. Second, *auditability*: $\hat{B}_t$ is an interpretable monitoring statistic ("the agent currently believes it has a 22% chance of being right"), useful for logging and human oversight independent of the routing decision. Third, *statistical efficiency*: the posterior is learned from all trajectories, whereas a direct policy gradient sees the cost signal only at decision boundaries. The price is model misspecification risk in the belief update, which Section 7 probes directly.

Theory versus production instantiation. The explicit Bayesian filter is the analyzable idealisation; in production we expect a *learned success predictor* $\hat{B}_t = f_\theta(\text{signal history})$ to replace it while preserving the decision structure. A learned predictor handles correlated, non-stationary signals that violate Assumption 1 and can ingest hidden-state features directly. The division of labour under this swap is clean: the regret analysis (Section 5.4) is filter-agnostic and becomes the *contract* the learned predictor must satisfy—minimise Brier score, verify calibration, then trust the thresholds—while the threshold-structure guarantee (Theorem 2) is what is formally lost, since an arbitrary learned predictor need not inherit the FOSD transition. Pragmatically one thresholds anyway, computing the schedule by running Algorithm 2 on *empirical* belief transitions from development trajectories rather than the analytic ones. We caution against going one step further and learning the escalate/continue policy end-to-end: that forfeits cost modularity (price changes then require retraining rather than a cheap backward induction) and the auditability of $\hat{B}_t$ as a monitoring statistic.

Overhead and task adaptivity. At inference the update is an $O(1)$ table lookup (or tiny MLP evaluation) per token, negligible next to a transformer forward pass; all expensive work (fitting, calibration, backward induction) is offline. The framework also adapts across task types without refitting the signal model: per-domain priors π_d and per-domain costs yield per-domain threshold schedules from the same fitted λ, again via Algorithm 2 alone.

7 Simulation Study

We call this a *simulation study* rather than an experiment, and we are candid about what it can and cannot show. Because we specify the data-generating process, this is a world in which the modelling

assumptions hold by construction; it is a check that the *derived policy behaves as the theory predicts*, and a diagnostic of how it degrades when an assumption is violated. It is emphatically not evidence about real LLM token dynamics—that is the role of the protocol in Section 9. All numbers come from a single reproducible script (fixed seed).

Setup. We draw $Y \sim \text{Bernoulli}(\pi)$ with $\pi = 0.60$ and, per token ($T = 40$), a signal $e_t \sim \text{Beta}(2, 4)$ if $Y = 1$ and $e_t \sim \text{Beta}(4, 2)$ if $Y = 0$ (the signal enters the update only through its likelihood ratio, so no shape assumption is needed; Lemma 2). The senior succeeds with $q = 0.90$. Costs are $L = 1$, $\kappa = 0.002$ per token, $\gamma = 0.15$. We evaluate on $N = 40{,}000$ queries. We compare: **junior-only**; **senior-only**; **fixed-rule** (escalate at the first token with $e_t > \theta$); **selective** (generate fully, escalate if final confidence $< \tau$, a post-hoc baseline that always pays full local generation); **Bayesian myopic schedule** (the literal, parameter-free rule of Eq. (4)); **Bayesian constant threshold** (escalate at the first token with $B_t < \tau$ for a constant τ, swept); and **Bayesian optimal-stopping** (Algorithm 2, parameter-free). Cost per query counts junior tokens plus escalation; accuracy is the fraction of correct final answers. *Disclosure:* the fixed-rule, selective, and constant-threshold policies each have one free parameter, which is swept and reported at the compute-matched point; the myopic-schedule and optimal-stopping rows involve no tuning.

Results. Figure 1 shows the cost–accuracy frontier. The Bayesian frontier dominates both baselines: for any compute budget it attains higher accuracy, and the optimal-stopping operating point (star) sits above and to the left of unconditional escalation. Table 1 reports a matched-compute slice at ≈ 0.11 cost/query. The Bayesian policy reaches 96.0% accuracy while escalating on only 40% of queries, versus 91.0% for the fixed rule and 90.1% for always escalating to the senior; it also matches the accuracy of the post-hoc selective baseline at lower cost, because it can abort early rather than always completing the local generation. That the number nominally exceeds senior-only accuracy should be given no weight: it is an artifact of the constant-q assumption (the senior's $q = 0.90$ does not degrade with query difficulty, so keeping the junior's confidently-correct easy cases mechanically lifts the mixture). With a realistic difficulty-dependent $q(x)$ the effect shrinks and can vanish. Throughout, the meaningful comparison is the *ordering and spacing of policies at matched cost*, not any absolute margin over the senior; Section 10 returns to this.

Two further honest readings of Table 1. First, the *literal* myopic schedule performs poorly: its first-token threshold $\tau_1^{\text{myo}} = 0.828$ exceeds the prior $\pi = 0.60$, so it escalates 62% of queries at the very first token (68% overall), reaching only 0.934 accuracy at *higher* cost (0.129)—a concrete demonstration of the early over-escalation predicted by Remark 4. The strong "Bayesian, constant threshold" row is a *tuned* rule, not Proposition 1; the untuned policy that performs well is the optimal-stopping schedule. Second, the matched-compute slice flatters the Bayesian–fixed-rule gap: at slightly higher compute the fixed rule nearly catches up ($\theta = 0.85$ gives 0.956 accuracy at cost 0.118). The robust claim is that the Bayesian frontier weakly dominates everywhere (Figure 1); the size of the point gap depends on where the budget lands on the fixed rule's steep region.

Belief dynamics and the threshold schedule. Figure 2 plots posterior trajectories: beliefs for eventual successes drift up and for eventual failures drift down, so a threshold cleanly separates them within a few tokens. The computed schedule illustrates Remark 4's small-κ regime: interior thresholds stay low (rising gently from 0.02 to 0.08 across the generation)—because escalating one step later costs only $\kappa = 0.002$ while buying another observation, the option value of waiting keeps interior escalation conservative—and the threshold jumps to the analytic terminal value $q - \gamma/L = 0.75$ exactly at the horizon, where no further information can arrive. The myopic

Table 1: Matched-compute comparison (≈ 0.11 cost/query) on the simulation model. Accuracy is fraction correct; "esc." is escalation rate. Rows marked "tuned" sweep one operating parameter and are reported at the compute-matched point; the myopic-schedule and optimal-stopping rows are parameter-free. Numbers are produced by the accompanying simulation.

Policy	Accuracy	Compute/query	Esc. rate
Junior only	0.601	0.080	0.00
Senior only	0.901	0.150	1.00
Fixed-rule (entropy, tuned)	0.910	0.114	—
Selective (post-hoc, tuned)	0.959	0.140	—
Bayesian, myopic schedule (Eq. (4))	0.934	0.129	0.68
Bayesian, constant threshold (tuned)	0.958	0.111	—
Bayesian, optimal-stopping	**0.960**	**0.111**	0.40

schedule (4), by contrast, is *highest* early; in this regime it over-escalates at the start of generation.

Calibration sensitivity. To probe Proposition 3 we contaminate the stream with "confidently wrong" queries: a fraction of $Y = 0$ cases whose signals are drawn from the success distribution. Figure 3 shows accuracy falling from 95.2% at 0% contamination to 85.7% at 30%, while the escalation rate *drops* (from 0.47 to 0.37) because the contaminated cases look confident and are wrongly kept local. This is exactly the $O(L)$ regret event of Proposition 3 and underlines that belief calibration, not the decision rule, is the binding constraint.

Sample-complexity check. To test Theorem 3's driver, we discretise the signal into $K = 20$ bins, fit the plug-in $\hat{\lambda}$ from n labelled trajectories (add-one smoothing), and compare against the exact binned oracle on a fixed evaluation set, averaging over 30 calibration resamples per n. The belief error $\mathbb{E}|\hat{B}_{10} - B_{10}|$ falls from 2.9×10^{-3} at $n = 25$ to 2.1×10^{-4} at $n = 3200$, with log–log slope -0.53—matching the predicted $1/\sqrt{n}$ rate. The realised end-to-end cost gap is already below 0.0016 (about 1% of total cost) at $n = 25$ and sits far under the bound at every n: with a well-separated signal, beliefs rarely linger near the threshold, so estimation errors rarely flip decisions. This is the theorem's pessimism working as intended—the bound is a worst-case guarantee, and the practical message is that modest labelled sets suffice when the signal is informative.

Robustness across observation models. To check that the conclusions are not an artifact of the Beta observation family, we repeat the comparison under two further signal models (Table 2): (i) *Gaussian* class-conditionals with equal variance, conditionally i.i.d.; and (ii) an *AR(1)-correlated* Gaussian model with the same marginals but lag-one correlation $\varphi = 0.6$, evaluated while the belief update *still assumes independence*—a deliberate violation of Assumption 1 that makes the update overconfident (it double-counts correlated evidence). For each model we sweep each policy's operating parameter and report its best total-cost point. The Bayesian rule attains lower total cost than the tuned fixed rule in all three models. Under misspecification its edge narrows but does not invert (total cost 0.158 vs. 0.166), and accuracy degrades gracefully (95.6% vs. 95.8% in the matched i.i.d. model): correlation costs performance, consistent with the calibration analysis, but does not break the method.

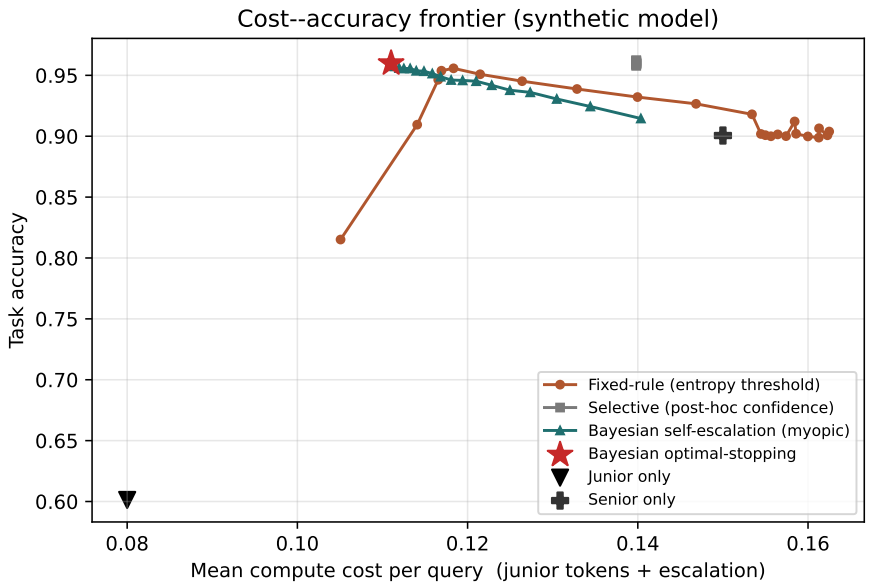

Figure 1: Cost–accuracy frontier on the simulation model. The Bayesian policies (teal) dominate the fixed-rule (orange) and post-hoc selective (grey) baselines; the optimal-stopping point (red star) beats unconditional escalation on both axes.

8 Real-Model Validation

The simulation of Section 7 verifies the derived policy in a world where the modelling assumptions hold by construction. Version 1 of this paper deferred any real-model evaluation, reporting only the pre-registered protocol of Section 9. This section, added in Version 1.1, takes a first, deliberately narrow step toward that protocol and runs the framework on a real hierarchical system. We are candid about scope: this is a single code-generation cascade, one model pair, greedy decoding, a single seed, and—importantly—a *constant-threshold* instantiation with a belief fit from running signal summaries rather than the recursive filter of Lemma 1. It is evidence that the competence signal and the escalation frontier behave as the theory predicts; it is *not* a test of the optimal-stopping dynamic program, which we leave to future work (Section 10). All numbers come from reproducible scripts with fixed seeds.

Setup. The junior is `Qwen2.5-Coder-1.5B-Instruct` and the senior `Qwen2.5-Coder-7B-Instruct`, both served locally with token-level log-probabilities. We evaluate on the sanitized MBPP test split (257 tasks), defining Y by execution against the task's unit tests—an unambiguous correctness label. For each greedy junior generation we log per-step token entropy, next-token log-probability, and top-2 margin. The competence posterior B_t is a logistic regression on the running (cumulative) means of these three signals, evaluated with 5-fold cross-validation so that every reported belief is out-of-fold. The senior attempts every task once (greedy), giving the counterfactual needed to price escalation.

Capability gap and escalation ceiling. The junior solves 62.3% of tasks and the senior 80.9%. Of the 97 junior failures the senior rescues 54 (55.7%); the remaining 43 (44.3%) are failed by both models and form an irreducible floor no routing policy can cross. The escalation ceiling— junior

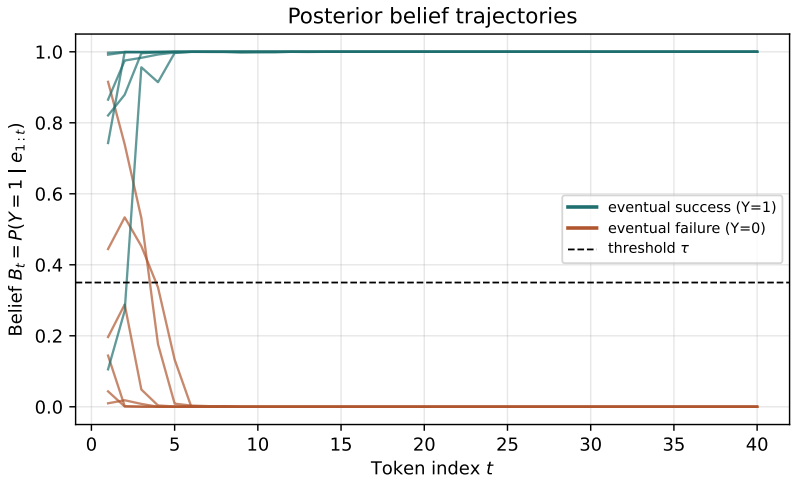

Figure 2: Posterior success-belief trajectories B_t for twelve queries. Eventual successes (teal) separate from eventual failures (orange) within a few tokens; the dashed line is an illustrative threshold.

Table 2: Robustness across observation models. Each policy is tuned to its best total-cost operating point per model; total cost = compute + expected error cost. AR(1) uses the i.i.d. belief update on correlated signals (misspecified). Numbers produced by the accompanying robustness script.

Observation model	Policy	Accuracy	Compute	Total cost
Beta, i.i.d. (baseline)	Fixed-rule	0.956	0.117	0.161
	Bayesian myopic	0.960	0.111	**0.151**
Gaussian, i.i.d.	Fixed-rule	0.958	0.117	0.160
	Bayesian myopic	0.958	0.111	**0.153**
Gaussian AR(1), φ=0.6 (misspecified update)	Fixed-rule	0.954	0.120	0.166
	Bayesian myopic	0.956	0.114	**0.158**

successes plus every failure escalated—is therefore 214/257 = 83.3%, not the senior's marginal 80.9%. This shared-failure floor is the real-model face of the constant-q caveat in Section 7: senior reliability is difficulty-correlated, so the achievable gain is bounded well below perfect rescue.

The competence signal is informative but imperfect. The cross-validated posterior attains AUROC 0.758 against eventual success. The per-step signal separates the classes early—mean entropy on eventual failures exceeds that on successes from $\sim 5\%$ of the generation—but *non-monotonically*: through the 25–55% band the instantaneous gap collapses and briefly inverts, precisely where confidently-wrong failures (low-entropy, incorrect) coincide with hard-but-correct successes (high-entropy, correct). This is the mechanism of Remark 1 and Proposition 3 observed directly: raw entropy is not the posterior, and confident errors are the binding failure mode.

Cumulative belief discrimination rises in t (prediction (b)). Although the *instantaneous* signal is non-monotonic, the *cumulative* posterior B_t is not: its discrimination increases near-monotonically over the generation (Spearman $\rho = 0.93$ between generation fraction and AUROC

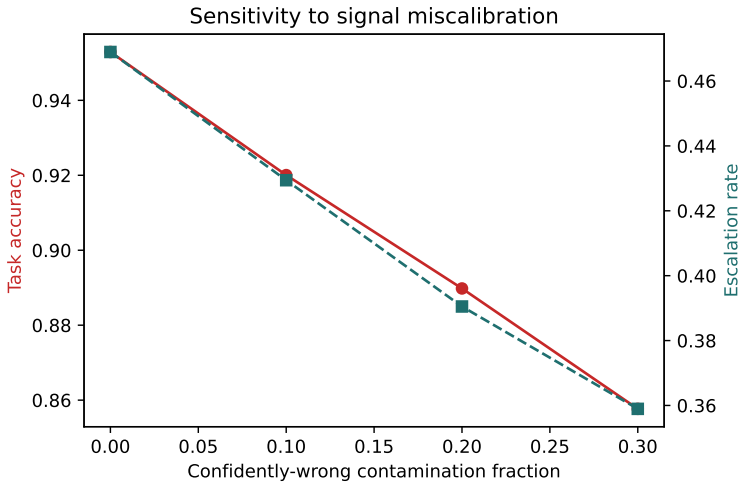

Figure 3: Sensitivity to signal miscalibration. As confidently-wrong contamination grows, accuracy degrades and—counter-productively—the escalation rate falls, because miscalibrated confidence suppresses the very escalations that are needed.

of B_t; AUROC rising from 0.51 to 0.76). Through the middle band where the instantaneous signal collapses, B_t *plateaus* rather than declining— confidently-wrong tokens stop contributing fresh evidence, but the belief retains what it accumulated earlier—and resumes rising thereafter (Figure 5). This confirms pre-registered prediction (b) of Section 9 and is direct support for the paper's central design choice: accumulate a calibrated belief rather than react to per-step uncertainty. We note the corresponding tension for early action: discrimination peaks at the horizon (t=T), so any policy that stops early necessarily acts on a weaker-than-terminal belief.

The escalation frontier dominates post-hoc routing (prediction (a)). We compare a *streaming* policy—escalate at the first step where the running belief falls below a threshold τ, aborting the remaining junior generation—against *post-hoc routing*, which runs the junior to completion and escalates the least-confident fraction (the confidence-cascade family the protocol names as a baseline). Sweeping each policy's operating parameter traces the cost–accuracy frontier of Figure 4. Streaming dominates post-hoc across the frontier: to reach 75% accuracy it uses 30.2% less total compute (14,841 vs. 21,273 generated tokens). More strikingly, streaming reaches 75% accuracy—+12.7 points over the junior alone—at essentially the junior's own compute (0.98× junior-only tokens): the tokens saved by aborting doomed generations offset the senior calls added. At τ=0.5 the policy escalates 37% of tasks, catching them at a mean of 29% of the way through generation, for 74.7% accuracy. The advantage over post-hoc is structural: post-hoc pays every junior generation in full before it can route, whereas streaming stops paying for a generation the moment the belief turns against it.

What this does and does not establish. Two of the three pre-registered predictions are supported on real data: the Bayesian frontier dominates the post-hoc baseline at equal cost (a), and the cumulative belief's AUROC rises in t (b). Prediction (c)—that the accuracy gap over baselines shrinks as calibration error grows—we do not test here; the simulation's calibration-sensitivity study (Figure 3) is its controlled analogue. Three limits bound the reading. First, the policy evaluated is a *constant* threshold, not the optimal-stopping schedule of Eqs. (5)–(6); by

Table 3: Real-model comparison on MBPP (sanitized test, 257 tasks; `Qwen2.5-Coder` 1.5B→7B). Compute is total generated tokens, normalised to junior-only. Post-hoc and streaming are reported at the operating point reaching 75% accuracy; the belief is cross-validated. Numbers produced by the accompanying harvest and analysis scripts.

Policy	Accuracy	Compute (× junior)	Esc. rate
Junior only	0.623	1.00	0.00
Senior only	0.809	—	1.00
Post-hoc routing (tuned)	0.750	1.41	—
Streaming (tuned τ)	0.747	**0.98**	0.37

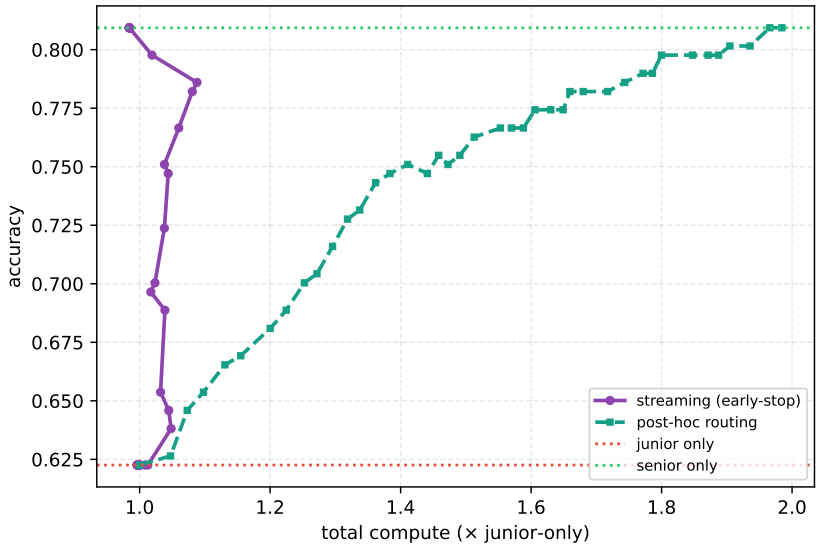

Figure 4: Real-model cost–accuracy frontier. Streaming escalation (purple) dominates post-hoc routing (teal) everywhere: it climbs from junior to senior accuracy at near-constant compute, because aborting doomed generations offsets the added senior calls. Dotted lines mark junior-only and senior-only accuracy.

Theorem 2 a threshold rule is the right *form*, but the option-value machinery—the contribution over classical sequential testing—is not exercised, and the empirical margin of the dynamic program over the best constant threshold remains to be measured. Second, the belief is a running-mean proxy for the recursive filter of Lemma 1, as anticipated in Section 6. Third, this is one benchmark, one model pair, greedy decoding, and a single seed; the protocol's reasoning and commonsense datasets, and the sampling-based baselines, remain open. Within those limits, the framework transfers: the signal is informative, the cumulative belief behaves as predicted, and confidence-routed escalation is markedly more compute-efficient than routing after the fact.

9 Protocol for Real LLM Systems

We pre-registered the full evaluation below to make the empirical test falsifiable; it was specified in Version 1 of this paper as a plan, before any real-model run. Section 8, added in Version 1.1,

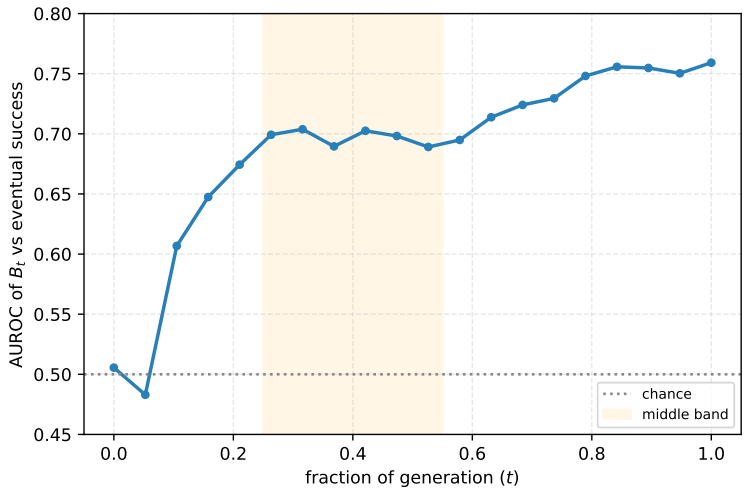

Figure 5: Discrimination of the cumulative belief B_t versus generation fraction (AUROC of B_t against eventual success, cross-validated). The curve rises near-monotonically (Spearman $\rho = 0.93$) and *plateaus*—rather than declining—through the shaded 25–55% band where the instantaneous signal collapses, confirming prediction (b).

reports its first execution on a code-generation cascade—keeping prediction and test separated in time—and finds two of the three predictions confirmed.

Systems. Junior: a small instruction/coding model served with token-level logit access (so that entropy and next-token margin are available at each step). Senior: a substantially stronger reasoning/coding model. Both served through an inference stack that exposes per-token log-probabilities.

Signals. Per-step e_t candidates: token entropy, next-token probability margin, and a single-pass semantic-entropy probe [11]. Each is calibrated separately so that comparisons isolate the signal's quality.

Belief fitting and calibration. On a labelled development split, run the junior, record $(e_{1:T}, Y)$, fit λ, and calibrate B_t by isotonic regression. Report reliability diagrams and expected calibration error (ECE) for B_t at several t.

Baselines. (i) junior-only; (ii) senior-only; (iii) query-level router [19]; (iv) confidence cascade [8]; (v) sampling-based semantic-entropy deferral [3]; (vi) our myopic and optimal-stopping policies.

Datasets. A reasoning set (e.g. multi-hop QA), a commonsense set with natural easy/hard structure, and a code set with executable unit tests to define Y unambiguously; plus an in-domain deployment set.

Metrics. Cost–accuracy Pareto frontier (primary); escalation precision/recall; calibration (ECE, Brier, AUROC of B_t vs. Y); and decision/end-to-end latency, including the cost of computing e_t.

Falsifiable predictions. If the framework transfers, then (a) the Bayesian policy's frontier should dominate the query-level router and the confidence cascade at equal cost; (b) AUROC of B_t should rise monotonically in t; and (c) the accuracy gap over baselines should shrink as calibration error grows, per Proposition 3. Failure of (a)–(c) would falsify the central claims.

10 Limitations

Modelling assumptions. Assumption 1 (conditional-i.i.d. signals) is false for real token streams, which are correlated and non-stationary; the update then becomes an approximation and λ should be replaced by a sequence model of the signal. (No shape assumption on the signal densities is needed for the threshold structure itself; Lemma 2 holds for any density pair.)

Confidently-wrong predictions. As Proposition 3 and Figure 3 show, the method inherits the calibration of its signal. Where the junior is confidently wrong, no threshold policy on that signal can help; combining epistemic signals with lightweight cross-model agreement [10] is a natural remedy.

Logit access. Token-level signals require an inference stack that exposes log-probabilities; many managed APIs return them only after generation, or not at all, restricting deployment to self-hosted or logprob-exposing endpoints.

Senior independence. We treat q as constant; in practice senior success correlates with query difficulty, and a difficulty-conditioned $q(x)$ would tighten the decision.

Scope of the real-model study. Version 1 flagged the absence of real-model results as its central limitation; Version 1.1 partially closes it. Section 8 validates the framework on a single code-generation cascade, one model pair, greedy decoding, and a single seed. Three limits bound it: the policy evaluated is a *constant* threshold, not the optimal-stopping schedule of Eqs. (5)–(6), so the option-value contribution over classical sequential testing is not yet exercised empirically; the belief is a running-mean proxy for the recursive filter of Lemma 1; and prediction (c) of Section 9 remains untested on real models. The remaining datasets and baselines of the protocol are open.

11 Conclusion

We framed the question of *when an agent should ask for help* as Bayesian self-escalation: a junior model tracks an online posterior over its own eventual success from the uncertainty signals it emits and defers to a stronger model when the expected utility of deferral wins. The framework yields a closed-form myopic threshold, an optimal-stopping characterisation with a proven threshold structure, and a regret bound that pins the method's success to belief calibration. A controlled simulation study confirms the predicted behaviour, including the myopic–optimal threshold gap and the calibration-driven failure mode. The decisive next step is the real-system protocol of Section 9; whether token-level signals on current LLMs are calibrated enough to realise these gains is, in the end, an empirical question this paper is designed to make testable.

Reproducibility. The simulation study (Section 7) is generated by three self-contained scripts (main simulation, robustness study, and sample-complexity experiment); the real-model validation (Section 8) by a harvest script and three post-hoc analysis scripts (separation and frontier, streaming escalation, and the B_t AUROC test). All are released with this paper. The paper and simulation scripts are permanently archived at DOI: 10.5281/zenodo.21330787; the full code, including the real-model pipeline, is at `github.com/nadeem-shaikh/llm-self-escalation`.

References

[1] L. Chen, M. Zaharia, and J. Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv:2305.05176*, 2023.

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

[3] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024.

[4] T. S. Ferguson. *Optimal Stopping and Applications*. Electronic text, UCLA, 2006.

[5] Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. In *NeurIPS*, 2017.

[6] Y. Geifman and R. El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *ICML*, 2019.

[7] A. Graves. Adaptive computation time for recurrent neural networks. *arXiv:1603.08983*, 2016.

[8] W. Jitkrittum, N. Gupta, A. K. Menon, H. Narasimhan, A. Rawat, and S. Kumar. When does confidence-based cascade deferral suffice? In *NeurIPS*, 2023.

[9] S. Kadavath, T. Conerly, A. Askell, et al. Language models (mostly) know what they know. *arXiv:2207.05221*, 2022.

[10] S. Kolawole, D. Dennis, A. Talwalkar, and V. Smith. Agreement-based cascading for efficient inference. *Transactions on Machine Learning Research*, 2025.

[11] J. Kossen, J. Han, M. Razzak, L. Schut, S. Malik, and Y. Gal. Semantic entropy probes: Robust and cheap hallucination detection in LLMs. *arXiv:2406.15927*, 2024.

[12] V. Krishnamurthy. *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*. Cambridge University Press, 2016.

[13] L. Kuhn, Y. Gal, and S. Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *ICLR*, 2023.

[14] Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023.

[15] W. S. Lovejoy. Some monotonicity results for partially observed Markov decision processes. *Operations Research*, 35(5):736–743, 1987.

[16] A. Madaan, N. Tandon, P. Gupta, et al. Self-Refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

[17] D. Madras, T. Pitassi, and R. Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *NeurIPS*, 2018.

[18] H. Mozannar and D. Sontag. Consistent estimators for learning to defer to an expert. In *ICML*, 2020.

[19] I. Ong, A. Almahairi, V. Wu, W.-L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, and I. Stoica. RouteLLM: Learning to route LLMs with preference data. *arXiv:2406.18665*, 2024.

[20] G. Peskir and A. Shiryaev. *Optimal Stopping and Free-Boundary Problems*. Birkhäuser, 2006.

[21] T. Schuster, A. Fisch, J. Gupta, M. Dehghani, D. Bahri, V. Q. Tran, Y. Tay, and D. Metzler. Confident adaptive language modeling. In *NeurIPS*, 2022.

[22] C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv:2408.03314*, 2024.

[23] D. Soiffer, S. Kolawole, and V. Smith. Semantic agreement enables efficient open-ended LLM cascades. In *EMNLP (Industry Track)*, 2025.

[24] R. Verma and E. Nalisnick. Calibrated learning to defer with one-vs-all classifiers. In *ICML*, 2022.

[25] A. Wald. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[26] A. Wald and J. Wolfowitz. Optimum character of the sequential probability ratio test. *Annals of Mathematical Statistics*, 19(3):326–339, 1948.