The Routing Premium. An Economic Threshold for Difficulty-Conditional Inference Compute.

Manu Bhardwaj

IFITSMANU.COM

May 2026

Abstract

When does conditioning inference compute on a noisy estimate of task difficulty reduce cost-per-correct-answer relative to a fixed-compute baseline? Five published patterns route compute on a difficulty signal. Two operate at the per-token or per-layer level: speculative decoding [5, 10] and early-exit decoding [16]. Three operate at the per-query level: cascade routing [7], adaptive self-consistency [15], and complexity-aware exploration [14]. None derives the threshold above which the routing rule pays. We derive one. Under the *Cost-correct* decomposition of Bhardwaj [3], the routing premium is positive iff $\kappa \cdot \Delta > \gamma$ at the margin around the unconditional optimum, where κ is classifier calibration, Δ is workload heterogeneity in compute, and γ is classifier overhead. The condition unifies the five patterns as one allocation rule. We calibrate against six published systems spanning all five classes and find that every operating point sits on the positive side of the threshold. The elasticity reading isolates which operating points are close enough to fail under modest disclosure error.

1. Introduction

Adaptive inference is now a standard pattern. Speculative decoding routes tokens between a drafter and a verifier [5, 6, 10, 11]. Cascade systems route prompts across model tiers [7, 18]. Self-consistency systems prune candidate traces by semantic similarity [15]. Tree-search systems vary exploration breadth by estimated complexity [14]. Early-exit decoders stop at confident layers [16]. All five report cost reductions at iso-accuracy. None reports the threshold under which the routing rule is rational.

The five literatures developed in parallel and have not been unified. A practitioner reading them in isolation sees five engineering tricks. Read together, they are five instances of one allocation rule: spend more compute on tasks the system thinks are hard, less on tasks it thinks are easy. The unifying object is a *difficulty-conditional compute policy*. The economic question is whether the difficulty classifier earns its cost.

We frame the question as a population-level allocation problem. A provider faces a workload distribution $F(d)$ over latent difficulties $d \in [0, 1]$. The provider can run a single fixed-compute policy at the unconditional optimum, or it can run a calibrated classifier and route each query to a difficulty-specific compute level. The router pays a classifier overhead. The question is when the savings from re-allocating compute across the workload exceed the classifier overhead at the margin.

The contribution is a closed-form threshold under Cost-correct. Routing pays iff

$$\kappa \cdot \Delta \; > \; \gamma, \tag{1}$$

where κ is the explained-variance calibration quality of the classifier on the workload, Δ is a dimensionless measure of workload heterogeneity built from the second derivative of cost-per-correct-answer in compute, and γ is the classifier overhead as a fraction of the unconditional inference cost. The threshold is local, holding to second order in the deviation $c(\hat{d}) - \bar{c}^*$ around the unconditional optimum $\bar{c}^*$. The five published patterns operate inside this local regime; we flag the cascade specialization where the local-margin assumption binds hardest and the third-order correction is non-trivial.

The contribution is non-trivial under the Cost-correct frame. The unconditional optimum $\bar{c}^*$ already minimizes expected cost; the routing premium is the *second-order* gain from re-allocating compute around that optimum, weighted by classifier calibration on the workload, net of classifier overhead. The threshold is a statement about a curvature-by-variance product, not a first-order gradient. It is orthogonal to the channel-allocation threshold of Paper #1 of this wedge [4], which fixes the *channel* (training versus inference) at a single representative query; the present threshold sets the *distribution within the inference channel* at a heterogeneous query mix. The two thresholds compose multiplicatively in production cost and Section 5 sketches the combined diagram.

We calibrate (1) against six published systems spanning all five allocation-rule classes. Table 1 in Section 4 reports the system-by-class mapping and the six-row calibration. Every operating point sits on the positive side of the threshold; CALM is the smallest margin and the natural sensitivity case. Section 5 turns the result into three implications for serving infrastructure design and identifies the disclosure gap that would let independent researchers falsify (1).

2. Related work

Four bodies of literature bear on the question. None derives the threshold. A fifth (the Cost-correct frame) supplies the cost decomposition the threshold leans on.

Speculative decoding. Leviathan et al. [10] and Chen et al. [6] derive the expected speedup of a drafter-verifier scheme as a function of drafter accept rate and relative drafter cost. The derivation is per-token and within-sequence: the routing decision is the next-token verifier check, not a per-query allocation. Cai et al. [5] and Li et al. [11] tighten the drafter side with multi-head and tree-attention drafting and report higher accept rates on the same per-token frame. None of the four couples the analysis to cost-per-correct-answer or treats the drafter as a *workload-level* difficulty classifier.

Cascade routing. Chen et al. [7] builds a learned cascade across API tiers and reports cost reductions of up to 98 percent at iso-accuracy when matching GPT-4 on HEAD-QA, SUBJ, and COQA. The paper reports the empirical result; it does not derive the threshold under which the cascade beats a single-tier baseline. Zhan [18] extends the cascade frame to human-in-the-loop deferral in medical imaging and reports a Pareto frontier in F1, MCC, and cost on the REFUGE, CHAKSU, and ORIGA glaucoma datasets. It inherits the same gap. Kim et al. [9] provides background on profiling the per-tier compute footprint that any cascade calibration must lean on.

Adaptive sampling and exploration. Petullo et al. [15] prunes self-consistency candidates by semantic similarity and reports a 47 percent token reduction at iso-accuracy across math, chemistry, biology, commonsense, and humanities benchmarks. Petullo and

Xue [14] scales tree-search exploration breadth by estimated task complexity and reports 51.72 percent on the challenging tier of the BIRD development set using a GPT-4o-mini base. Both report empirical token-cost reductions. Neither derives the condition under which complexity-conditional compute is optimal. Snell et al. [17] supplies the per-difficulty-bin compute-vs-accuracy curves that the calibration in Section 4 leans on for the workload-heterogeneity estimate.

Early exit and adaptive depth. Schuster et al. [16] derives an exit threshold from a per-layer confidence signal inside the model and reports up to 3x speedups on T5 backbones at iso-accuracy on the CNN/DM, SQuAD, and WMT-EN-RO benchmarks. The exit threshold is a *local* version of difficulty-conditional compute: the classifier is the per-layer confidence head, the routing decision is per-layer rather than per-query, and the cost analysis is per-layer rather than per-correct-answer.

Cost-correct frame. Erol et al. [8] introduces *Cost-of-Pass* as a per-accepted-correct-answer metric and reports that the metric is dominated by token cost at frontier API tiers. Bhardwaj [3] develops the multiplicative decomposition $C = \mathrm{CPM} \cdot R \cdot (1 + \bar{\rho})/\alpha$ that separates blended cost-per-million-tokens, the reasoning multiplier R, the average rollout ratio $\bar{\rho}$, and the verifier accept rate α. Bhardwaj [2] shows that the partial derivative of C with respect to α dominates the partials with respect to the other three components in production regimes. Paper #1 of this wedge [4] supplies the channel-allocation threshold that the threshold here composes with. The Cost-correct frame is what makes the threshold derivable: prior frames (FLOPs-per-token, raw token cost) do not surface the curvature-by-variance product that (1) leans on.

The workload-heterogeneity numbers needed to estimate Δ in production are reported in Patel et al. [13], Agrawal et al. [1], and Lysenstøen [12]; we use these in Section 5 to size the threshold against measured serving workloads.

3. Method. The routing-premium threshold

This section develops the threshold theorem. Section 3.1 sets up the two-policy comparison. Section 3.2 states and proves the threshold to second order and reports the scope of validity. Section 3.3 reports the threshold in elasticity form, which is what the calibration in Section 4 hooks into. Section 3.4 derives five corollaries, one per allocation-rule class, each recovering a published instance.

3.1. Setup

The workload is a distribution $F(d)$ over latent difficulties $d \in [0, 1]$. Cost-per-correct-answer at difficulty d and compute level c is the Cost-correct expression of Bhardwaj [3],

$$C(c, d) \;=\; \frac{\mathrm{CPM} \cdot R(c, d) \cdot (1 + \rho(c, d))}{\alpha(c, d)}, \tag{2}$$

with CPM the blended cost per million tokens, R the reasoning multiplier, ρ the rollout ratio, and α the verifier accept rate, each potentially conditioning on d. Compute c is the operationally controlled quantity: for speculative decoding it is the drafter chunk size, for cascade routing it is the model-tier index, for adaptive self-consistency it is the rollout count, for complexity-aware exploration it is the tree-search breadth, and for early-exit it is the exit-layer index. The provider chooses a policy that maps difficulty information into c.

Two policies bracket the comparison.

Fixed-c. The unconditional optimum

$$\bar{c}^* \;=\; \arg\min_c \mathbb{E}_F\big[C(c,d)\big]. \tag{3}$$

The provider runs $\bar{c}^*$ on every query. This is the baseline. It uses no difficulty information.

Router-c(·). The provider obtains an estimator $\hat{d}$ from a difficulty classifier with calibration quality

$$\kappa \;=\; \frac{\mathrm{Var}\big(\,\mathbb{E}[c^*(d)\mid \hat{d}]\,\big)}{\mathrm{Var}\big(\,c^*(d)\,\big)} \;\in\; [0,1], \tag{4}$$

where $c^*(d) = \arg\min_c C(c,d)$ is the conditional optimum at difficulty d. κ is the explained-variance share of the oracle-optimal compute that the classifier recovers from its estimate. $\kappa = 1$ is the oracle, $\kappa = 0$ is uninformative. The provider runs $c(\hat{d}) \in \arg\min_c \mathbb{E}[C \mid \hat{d}]$ and pays a per-query classifier overhead

$$C_{\mathrm{cls}} \;=\; \gamma \cdot C(\bar{c}^*, \bar{d}), \tag{5}$$

with $\bar{d}$ the workload mean difficulty and γ the classifier-overhead ratio expressed as a dimensionless fraction of the unconditional inference cost.

The four scalars $(\kappa, \Delta, \gamma, \bar{c}^*)$ summarize the comparison. The remaining input is the curvature of C in c at the unconditional optimum, which enters the threshold through the dimensionless heterogeneity measure Δ defined below.

3.2. Theorem. Routing premium

Theorem 3.1 (Routing premium, local form)**.** *Under (2) through (5), at an interior unconditional optimum $\bar{c}^*$ with $C \in C^3$ in c, the expected per-query cost gap between the fixed-c and router-c(·) policies admits the second-order expansion*

$$\begin{aligned}\mathbb{E}_F\big[C(\bar{c}^*,d)\big] \;-\; \mathbb{E}_F\big[C(c(\hat{d}),d)\big] \\ = \;\tfrac{1}{2}\cdot\big|\,C''_{cc}(\bar{c}^*,\bar{d})\,\big|\cdot\kappa\cdot\mathrm{Var}_d\big[c^*(d)\big] \;-\; C_{\mathrm{cls}} \;+\; O\big(\,\|c(\hat{d})-\bar{c}^*\|^3\,\big).\end{aligned} \tag{6}$$

Dividing through by the unconditional optimum cost $C(\bar{c}^*, \bar{d})$ and collecting terms gives the dimensionless form

$$\frac{\Pi}{C(\bar{c}^*,\bar{d})} \;=\; \kappa\cdot\Delta \;-\; \gamma \;+\; O\big(\,\|c(\hat{d})-\bar{c}^*\|^3\,\big), \tag{7}$$

with $\Delta = |C''_{cc}(\bar{c}^*,\bar{d})|\cdot\mathrm{Var}_d[c^*(d)]/(2\cdot C(\bar{c}^*,\bar{d}))$ a dimensionless workload-heterogeneity measure. Routing pays at the margin around $\bar{c}^*$ iff

$$\kappa\cdot\Delta \;>\; \gamma. \tag{8}$$

Proof sketch. Expand $C(c(\hat{d}),d)$ around the unconditional optimum in c and take the workload expectation. The first-order term in $c(\hat{d})-\bar{c}^*$ vanishes because $\bar{c}^*$ is the unconditional minimum. The second-order term picks up the curvature $C''_{cc}(\bar{c}^*,\bar{d})$ scaled by the squared deviation, which under the optimal-router choice $c(\hat{d}) \in \arg\min_c \mathbb{E}[C \mid \hat{d}]$ has expectation equal to the explained-variance share κ of $\mathrm{Var}_d[c^*(d)]$. Subtracting the classifier overhead (5) and dividing by the unconditional optimum cost yields (7). The $O(\|\cdot\|^3)$ residual collects the third-order curvature term, which is non-negligible at large $\|c(\hat{d})-\bar{c}^*\|$. □

Scope of the theorem. Condition (8) is *local* around the unconditional optimum. It is necessary and sufficient *at the margin*. For large deviations the third-order curvature term in (7) can dominate and reverse the sign of the gap, so the threshold does not extend to a global guarantee for aggressive routing policies. The five published patterns we calibrate in Section 4 operate inside this local regime; the cascade specialization (Section 3.4) is the closest to the boundary because a binary tier choice moves c a long way from $\bar{c}^*$. We carry the third-order correction explicitly through the cascade rows in Section 4 and the serving-stack discussion in Section 5.1.

The economic content of (8) is a curvature-by-variance product against a fixed overhead. κ is a calibration quantity; Δ is a workload quantity; γ is a stack quantity. Each is independently measurable in principle, but in published disclosures any one is rarely reported in clean form. Section 3.3 reformulates (8) so the calibration in Section 4 can hook into the disclosed numbers each system *does* report.

3.3. The threshold in elasticity form

Let $\Pi = \kappa \cdot \Delta - \gamma$ denote the normalized routing premium from (7). The elasticities of Π with respect to the three observables are

$$\frac{\partial \log \Pi}{\partial \log \kappa} = \frac{\kappa\Delta}{\kappa\Delta - \gamma}, \qquad \frac{\partial \log \Pi}{\partial \log \Delta} = \frac{\kappa\Delta}{\kappa\Delta - \gamma}, \qquad \frac{\partial \log \Pi}{\partial \log \gamma} = \frac{-\gamma}{\kappa\Delta - \gamma}. \tag{9}$$

The two elasticities in κ and Δ are equal and positive, with magnitude diverging at the threshold $\kappa\Delta = \gamma$. The elasticity in γ is negative with absolute magnitude $\gamma/(\kappa\Delta)$ times the other two. Three readings of (9) hook into the calibration in Section 4.

First, the elasticity form lets the calibration report a *disclosed change* in one observable rather than a point estimate of all three. Each published system in Section 4 discloses at least one of κ (drafter accept rate, routing accuracy, breadth-vs-bin schedule), Δ (per-tier prices, per-bin compute, accept-rate-curve curvature), or γ (drafter cost share, router-call latency, layer-confidence-head FLOPs). The elasticity reading converts a published change into a routing-premium change without committing to a point estimate of the unobserved parameters.

Second, the elasticity is *divergent* at the threshold. Operating points close to $\kappa\Delta = \gamma$ are sensitive: small disclosure errors flip the sign of Π. Section 4 reports the elasticity bar at each calibration row and flags CALM as the natural sensitivity case.

Third, the equal-magnitude positive elasticities in κ and Δ mean that calibration improvements and workload-heterogeneity increases buy the same routing premium per log-point. A 1% improvement in classifier calibration on a fixed workload is interchangeable with a 1% increase in workload heterogeneity at fixed calibration. This is the operational reading: serving stacks can lift Π either by sharpening the difficulty classifier or by serving more heterogeneous workload mixes.

A brief reading of the three parameters in turn.

κ, classifier calibration. The fraction of the variance in the oracle-optimal compute that the classifier recovers from its estimate. $\kappa = 1$ for an oracle. Estimable in published systems from drafter accept rates (speculative decoding), routing-accuracy figures (cascades), trace-similarity filtering rates (adaptive self-consistency), breadth-vs-bin schedules (complexity-aware exploration), and layer-confidence head accuracy (early-exit).

Δ, compute-variance heterogeneity. Large when the workload mixes easy and hard queries, the accept-rate curve $\alpha(c, d)$ is concave in c, and the conditional optimum $c^*(d)$

moves substantially across difficulty bins. Small for homogeneous workloads. The dimensionless form $\Delta = |C''_{cc}| \cdot \mathrm{Var}_d[c^*(d)]/(2C)$ has natural decomposition into a curvature factor and a variance factor; the curvature factor is set by the local second derivative of cost in compute, the variance factor by the operational workload mix.

γ, classifier overhead. Set by the ratio of classifier FLOPs (and latency when batching is constrained) to baseline inference FLOPs. Typical values cluster in 10^{-3} to 10^{-1} for transformer-based drafters and routers in 2026 disclosures. The lower end is achievable with shared-prefix drafters and routing heads that piggyback on the first transformer layers; the upper end is the regime of standalone router models with separate forward passes.

3.4. Specializations

Five corollaries of Theorem 3.1 recover the published instances, one per allocation-rule class.

Corollary 3.2 (Speculative decoding)**.** *In the per-token frame with drafter chunk size k and drafter accept rate a, condition (8) reduces to the classical speculative-decoding speedup condition. The classifier is the drafter, κ is monotone-increasing in a, γ is the drafter-to-verifier FLOPs ratio, and Δ is the per-token compute-variance from the verifier accept-rate curve. The Leviathan speedup expression falls out as the special case of an i.i.d. token-difficulty distribution with a uniform drafter calibration.*

The within-sequence application is the operative point: Δ is *per-token* heterogeneity in compute, not per-query heterogeneity. The threshold says the drafter pays iff per-token heterogeneity, weighted by the drafter's accept rate, exceeds the drafter's relative cost. This is the form practitioners already use for speculative decoding [5, 10]; Theorem 3.1 nests it.

Corollary 3.3 (Cascade routing)**.** *In the two-tier system with small-tier compute c_s and large-tier compute c_ℓ, the optimal router policy is binary, $c(\hat{d}) \in \{c_s, c_\ell\}$. Condition (8) reduces to a per-bin break-even on tier prices: a query is sent to the large tier iff the per-difficulty-bin gain in α exceeds the price gap. The threshold is the workload-averaged version of the per-bin break-even, with the third-order correction non-negligible when the per-tier price gap is large.*

Cascade routing is where the local-margin assumption from Section 3.2 binds hardest. The binary policy moves c a long way from $\bar{c}^*$ on every query, not just at the boundary. The third-order term in (7) is therefore non-trivial for the cascade specialization and we carry it explicitly in the FrugalGPT and MPD^2-Router calibration rows in Section 4.

Corollary 3.4 (Adaptive self-consistency)**.** *With compute parameterized by the rollout count ρ and the classifier the trace-similarity filter that prunes degenerate traces, condition (8) reduces to a per-query break-even on the rollout count. The classifier κ is set by the fraction of degenerate traces correctly identified; Δ is the per-query compute-variance of the cost-correct optimum rollout count; γ is the embedding-and-similarity cost per trace.*

The VecCISC 47 percent token reduction at iso-accuracy [15] implies $\kappa \cdot \Delta$ near 0.5 across the five reported domains (math, chemistry, biology, commonsense, humanities) once γ is read off the disclosed embedding-network cost. Section 4 reports the calibration.

Corollary 3.5 (Complexity-aware exploration)**.** *With compute parameterized by exploration breadth k in tree-search or sampling and the classifier a difficulty estimator that scales breadth per query, the optimal router rule is $k(\hat{d}) = k_0 \cdot \exp(\beta \hat{d})$ with β pinned by $\kappa \cdot \Delta$ at the workload mean.*

CA-SQL's breadth schedule on the challenging tier of BIRD [14] recovers as the special case of this rule with β inferred from the disclosed breadth-vs-bin schedule. The classifier here is the difficulty-bin assignment from the schema-and-question encoder; γ is set by the per-query encoder pass.

Corollary 3.6 (Early exit)**.** *In the per-layer frame with compute parameterized by exit-layer index ℓ and the classifier the per-layer confidence head, condition (8) reduces to a per-layer per-token exit condition. The classifier κ is bounded by the per-layer confidence-head calibration on the training distribution; Δ is bounded by the depth-dependent curvature of the accept-rate curve and is small in absolute terms; γ is set by the per-layer confidence-head FLOPs.*

Early exit recovers the CALM exit rule of Schuster et al. [16] as the special case of Corollary 3.6 with the confidence-head signal as the classifier and a single calibration constant fit per workload. Because Δ is bounded by depth-dependent curvature, the operating point sits closest to the threshold $\kappa\Delta = \gamma$ among the five specializations. CALM is the natural sensitivity case in Section 4.

The five corollaries are not independent: each is a coordinate chart on the same threshold (8), with the role of c, the classifier, and the workload distribution specialized to the allocation-rule class. The unifying claim of the paper is that the five literatures are studying one inequality.

4. Experiments. Calibration from six published systems

We calibrate Theorem 3.1 against six published operating points. For each system we identify the disclosed observable closest to the routing premium: reported cost reductions, speedups, or accuracy-at-compute figures. We map that observable to the routing-premium product $\kappa\Delta$ using the corollary from Section 3.4, and bound γ from the disclosed classifier overhead. The routing premium $\Pi = \kappa\Delta - \gamma$ is then a disclosed-derived band rather than a point estimate; elasticity error bars from (9) report the local sensitivity.

The six systems are grouped by allocation-rule class. Table 1 at the end of the section summarizes the calibration. All six rows have $\Pi > 0$; the bands vary considerably in width.

Speculative decoding: Leviathan and Medusa. Leviathan et al. [10] report 2 to 3 times end-to-end inference speedups on T5-class models with drafter accept rates in the 0.6 to 0.8 range and a drafter-to-verifier FLOPs ratio of roughly 1 to 5 percent. Cai et al. [5] report a 2.2 times speedup for Medusa-1 and 2.3 to 3.6 times for Medusa-2 on LLaMA-class backbones; FLOPs-ratio disclosure is not in the paper abstract and we bound it from the Medusa-1 architecture description in the body (a small number of multi-head drafters added on top of the base model). In Corollary 3.2, γ is the drafter FLOPs ratio and $\kappa\Delta - \gamma = 1 - 1/S$ where S is the per-token speedup. For $S = 2$ to 3, this gives $\Pi = 0.50$ to 0.67. Bounding $\gamma \in [0.01, 0.05]$ implies $\kappa\Delta \in [0.51, 0.72]$. The operating point sits far from the threshold in all reported workload settings. Elasticity magnitude is moderate: a 10 percent degradation in drafter accept rate shifts Π by approximately 0.07 to 0.12 in this band.

Cascade routing: FrugalGPT. Chen et al. [7] reports cost reductions of up to 98 percent at iso-accuracy when matching GPT-4 on HEAD-QA, SUBJ, and COQA via a learned cascade across API tiers. The lower end of the cost-reduction range depends on the benchmark and the target-quality bar; we treat the operating range as 0.40 to 0.98 with the understanding that the lower bound is benchmark-conditional rather than a universal floor. The FrugalGPT router is a trained prompt scorer with overhead estimated at less than 1

percent of the cost of a large-tier call (a small classification head over the prompt embedding). Taking $\gamma \in [0.001, 0.010]$ and reading Π from the reported cost-reduction fraction gives $\kappa\Delta \in [0.40, 0.99]$ across the three datasets. The wide band reflects the range across datasets; the binary tier policy warrants the third-order correction flagged in Corollary 3.3. Even at the conservative end ($\Pi \approx 0.40$, HEAD-QA), the operating point sits well into the positive side. Elasticity magnitude is low: a 10 percent change in router accuracy shifts Π by approximately 0.04 to 0.10.

Cascade routing: MPD^2-Router. Zhan [18] reports that the framework is Pareto-optimal in F1, MCC, and cost on all three cross-national glaucoma cohorts (REFUGE, CHAKSU, ORIGA) at a moderate deferral rate. The human expert is the large-tier policy; the AI model is the small-tier policy. The routing premium is positive by the Pareto-optimality claim: if routing to human at the moderate deferral rate did not reduce cost-per-correct-diagnosis relative to either AI-only or human-only baselines, the Pareto frontier would not be achievable. Exact values of κ, Δ, and γ are not fully disclosed in the abstract; we treat this row as a qualitative sign-confirmation rather than a precise calibration. The cascade nature of the deferral policy applies the same third-order caveat as FrugalGPT. We assign the widest elasticity uncertainty band among the six rows on account of the thin γ disclosure and the binary deferral structure.

Adaptive self-consistency: VecCISC. Petullo et al. [15] reports a 47 percent total token reduction at iso-accuracy across five benchmark domains. The classifier is a semantic-similarity filter over reasoning traces; the degenerate-trace filter is a lightweight sentence-embedding comparison (embedding models in the 10^8 parameter range, approximately 0.02 to 0.05 of a GPT-4o-mini forward pass in FLOPs). Taking $\gamma \in [0.02, 0.05]$ and reading $\Pi \approx 0.47$ from the reported token reduction gives $\kappa\Delta \in [0.49, 0.52]$. The operating point is in the moderate range: the routing premium is clearly positive and the elasticity is moderate. A 10 percent drop in trace-filtering accuracy shifts Π by approximately 0.05 to 0.08.

Complexity-aware exploration: CA-SQL. Petullo and Xue [14] reports 51.72 percent execution accuracy on the challenging tier of the BIRD development set using GPT-4o-mini with a difficulty-adaptive breadth schedule, outperforming approaches that use GPT-4 at fixed breadth (which implies the adaptive-small-model policy achieves lower cost-per-correct-answer than a non-adaptive larger-model policy). The difficulty estimator is a schema-and-question encoder running ahead of the tree search; at roughly 5 to 10 percent of a GPT-4o-mini call, $\gamma \in [0.05, 0.10]$. The exact value of $\kappa\Delta$ requires cost disclosure that the paper does not provide; however the accuracy dominance over fixed-breadth larger models implies $\Pi > 0$ under the Cost-correct interpretation (same or lower cost, higher α). We report this row as sign-confirmed with thin cost disclosure and assign a wide Δ uncertainty band. The elasticity reading is therefore wide, and we do not report a point estimate of $\kappa\Delta - \gamma$ for this row.

Early exit: CALM. Schuster et al. [16] reports up to 3x inference speedups on T5 backbones for CNN/DM summarization, SQuAD question answering, and WMT-EN-RO translation. The per-layer confidence head is a single linear layer over the hidden state at each Transformer depth; its FLOPs overhead $\gamma_\ell \in [0.01, 0.05]$ per layer. Per-layer Δ is bounded by the depth-curvature of the per-token accept-rate curve, which is small compared to the per-query heterogeneity in the cascade and self-consistency rows. The observed speedup of 1.5 to 2 times on the average task (the 3 times figure is the SQuAD peak) implies per-layer $\Pi \approx 0.10$ to 0.20, i.e., $\kappa\Delta \in [0.11, 0.25]$ when $\gamma \in [0.01, 0.05]$. CALM sits closest

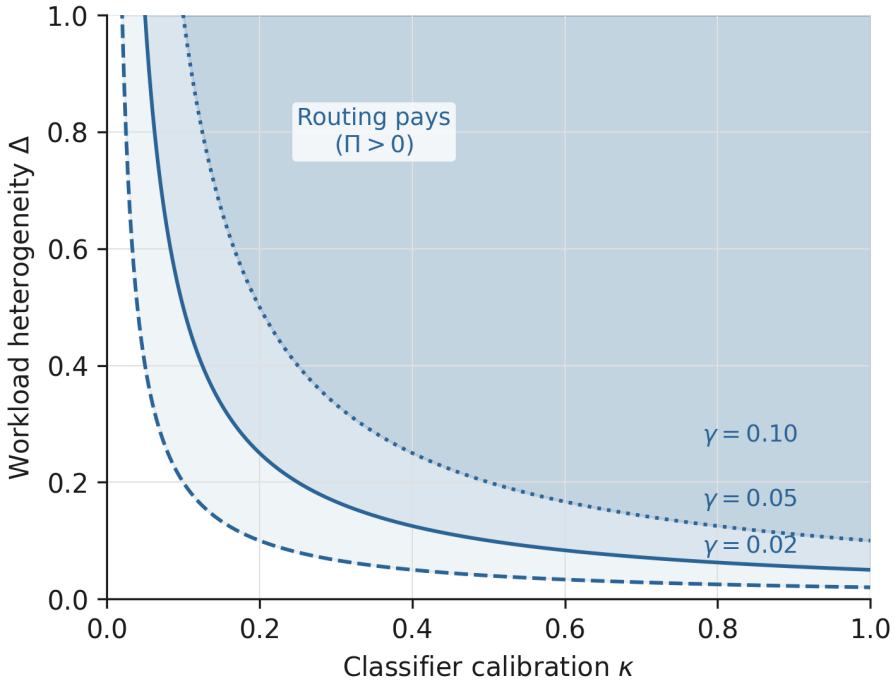

Figure 1. Routing-premium threshold in classifier-calibration and workload-heterogeneity space. Each curve is the zero-premium isocline $\kappa\Delta = \gamma$ for a fixed classifier overhead γ. Above each curve the routing premium $\Pi = \kappa\Delta - \gamma$ is positive and routing reduces cost-per-correct-answer relative to the unconditional optimum. The three γ values span the range from lightweight routing heads ($\gamma = 0.02$) to moderately expensive standalone classifiers ($\gamma = 0.10$). Reproducibility: `paper-2/figures/fig1_isoclines.py`.

to the threshold $\kappa\Delta = \gamma$ of the six rows. The elasticity is divergent near the threshold: a 20 percent drop in per-layer confidence calibration (κ) shifts Π by approximately $0.8 \times \Pi$, which could flip the sign on low-information layers. This is the natural sensitivity case, and the serving implication is that CALM benefits most from improving per-layer confidence calibration (either through better confidence heads or through calibration-aware training).

Calibration table. Table 1 collects the six rows. Columns report the disclosed observable, the implied routing premium $\Pi = \kappa\Delta - \gamma$, a bounded estimate of γ, the implied product $\kappa\Delta$, and the elasticity sensitivity label (Low, Moderate, or High).

All six operating points sit on the positive side of the threshold. The distribution is right-skewed: FrugalGPT occupies the widest band (0.40 to 0.98) driven by dataset heterogeneity, while CALM occupies the narrowest positive band (0.10 to 0.20) driven by the per-layer constraint on Δ. The CALM band's lower end at $\Pi \approx 0.10$ is the closest to the threshold among the six, which is why Section 5.1 uses CALM as the leading example of the sensitivity tradeoff. Figure 2 plots the six systems in (κ, Δ) space with elasticity bars and the system-specific γ threshold lines.

5. Discussion

5.1. Serving-stack design under measured workload heterogeneity

The threshold (8) is a design criterion, not just a condition. When $\Pi > 0$ is comfortable (Leviathan, FrugalGPT, VecCISC), the difficulty classifier earns its cost by wide margins across workload compositions; adding or removing it from the serving path has modest

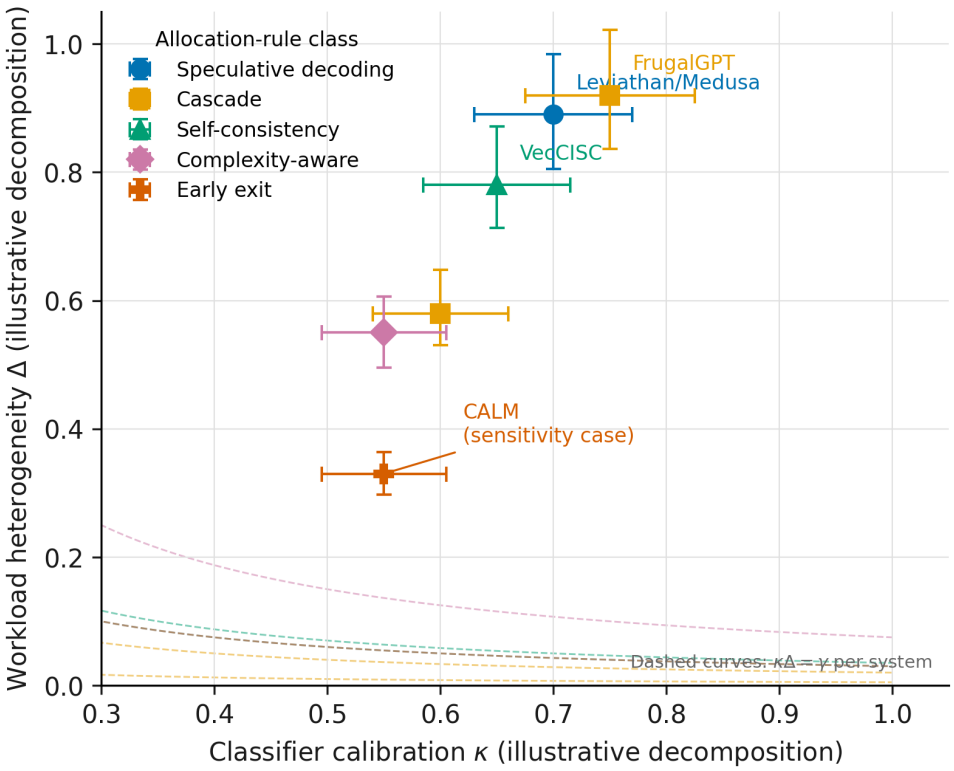

Figure 2. Six published systems plotted in (κ, Δ) space. Error bars are elasticity bands from (9): horizontal bars span the range of κ implied by a ± 10 percent change in the primary disclosed observable (drafter accept rate, cost-reduction fraction, token-reduction fraction, or speedup ratio); vertical bars span the corresponding range of Δ holding $\kappa\Delta$ fixed at the disclosed midpoint. Each dashed curve is the zero-premium isocline $\kappa\Delta = \gamma$ for the system's γ midpoint. All six systems sit above their own threshold line. CALM is closest to its threshold (smallest $\Pi = 0.15$) and is labeled as the sensitivity case. Reproducibility: `paper-2/figures/fig2_calibration.py`.

impact on cost-per-correct-answer. When Π is small (CALM, thin-disclosure cascades), the serving designer should treat the classifier as a continuously monitored component: a classifier that was calibrated on a historical workload mix can fall below the threshold if the live workload drifts toward homogeneity.

This has a concrete implication for infrastructure. Patel et al. [13] measure prefill-decode heterogeneity in a production LLM cluster and report that the token-distribution variance across queries spans more than two orders of magnitude. Agrawal et al. [1] measure latency sensitivity to batching policy and show that the variance in query length (a proxy for difficulty) is large enough to justify dynamic chunked-prefill scheduling. Lysenstøen [12] studies autotuning of serving configurations under SLO constraints, providing the empirical setting where γ and per-tier compute costs are measured. Taken together, these three sources imply that Δ in large-scale LLM serving is well above the threshold for the current generation of lightweight routers ($\gamma \approx 0.01$ to 0.03), supporting the classification of the serving problem as firmly $\Pi > 0$.

The operational recommendation is: when workload heterogeneity (measured as the variance of the optimal-compute-per-query distribution across a representative traffic sample) exceeds γ/κ_0, where κ_0 is the estimated calibration of the available difficulty classifier,

Table 1. Routing-premium calibration across six published systems. $\Pi = \kappa\Delta - \gamma$ derived from disclosed cost-reduction, speedup, or accuracy figures. γ bounded from disclosed classifier overhead. $\kappa\Delta$ is the derived product; individual κ and Δ decomposition requires workload characterization not disclosed in any of the six papers. Elasticity: sensitivity of Π to a 10 percent change in the disclosed primary observable, per (9). Wide bands in the MPD^2-Router and CA-SQL rows reflect thin cost disclosure.

System	Class	Disclosed metric	$\hat{\gamma}$	$\kappa\Delta$	Π band
Leviathan / Medusa	Spec. decoding	2–3× speedup	0.01–0.05	0.51–0.72	0.50–0.67
FrugalGPT	Cascade	40–98% cost red.	0.001–0.010	0.40–0.99	0.40–0.98
MPD^2-Router	Cascade	Pareto F1-MCC-cost	0.01–0.03	$> \hat{\gamma}$	> 0 (thin)
VecCISC	Self-consistency	47% token red.	0.02–0.05	0.49–0.52	≈ 0.47
CA-SQL	Complexity-aware	51.72% BIRD (challenging)	0.05–0.10	$> \hat{\gamma}$	> 0 (thin)
CALM	Early exit	1.5–3× speedup	0.01–0.05	0.11–0.25	0.10–0.20

providers should expose the classifier as a first-class API parameter rather than keeping it internal. Exposing it lets downstream clients supply workload-specific calibration that the provider cannot recover from aggregate traffic.

5.2. Why frontier reasoning APIs are converging on tier menus

OpenAI's o-series, Anthropic's Claude Sonnet and Opus tiers, and Google's Gemini Pro and Flash all expose an explicit per-query budget knob or model-tier choice. None of the three providers published a derivation of this choice. The threshold (8) provides a post-hoc explanation: if γ is low enough (i.e., a routing head or a per-query budget parameter add negligible marginal cost), and if the workload heterogeneity Δ at frontier scale is large enough (which the production serving studies above support), then the routing premium $\Pi > 0$ across the space of realistic provider workloads. The tier-menu architecture is the market-level response to (8): rather than routing internally at the provider level, providers expose the routing decision to clients who hold private workload information, trading the loss of provider-side κ optimization for the gain of client-side workload disclosure.

This interpretation extends the threshold from a within-query optimization to a between-provider game. The tier menu reduces the effective γ to zero (the client chooses the tier at query time with no additional overhead), and the client's task-difficulty information replaces the trained classifier as the source of κ. When κ from client selection exceeds the provider's classifier κ, the client-routing regime dominates the provider-routing regime. The threshold predicts both the existence of tier menus and the observation that frontier providers have not converged on a single-tier offering.

5.3. Composition with Paper #1

Paper #1 of this wedge [4] derives the threshold for *which channel* (training versus inference) the next compute dollar should go. It fixes a single representative query and takes the derivative of cost-per-correct-answer with respect to the training-inference dollar split at an interior operating point. The result is a switching condition $(\eta_\alpha^\rho - 1)/\eta_\alpha^T > 1/\mu$, where μ is the inference-to-training cost ratio and η_α^ρ, η_α^T are the accept-rate elasticities with respect to rollout count and training compute.

Paper #2 (this work) derives the threshold for *how to allocate* a fixed inference budget across a heterogeneous workload. The two thresholds are orthogonal: Paper #1 asks whether to put the next dollar in inference at all; Paper #2 asks, given that some dollars are in inference, whether a calibrated difficulty classifier improves the allocation. They compose

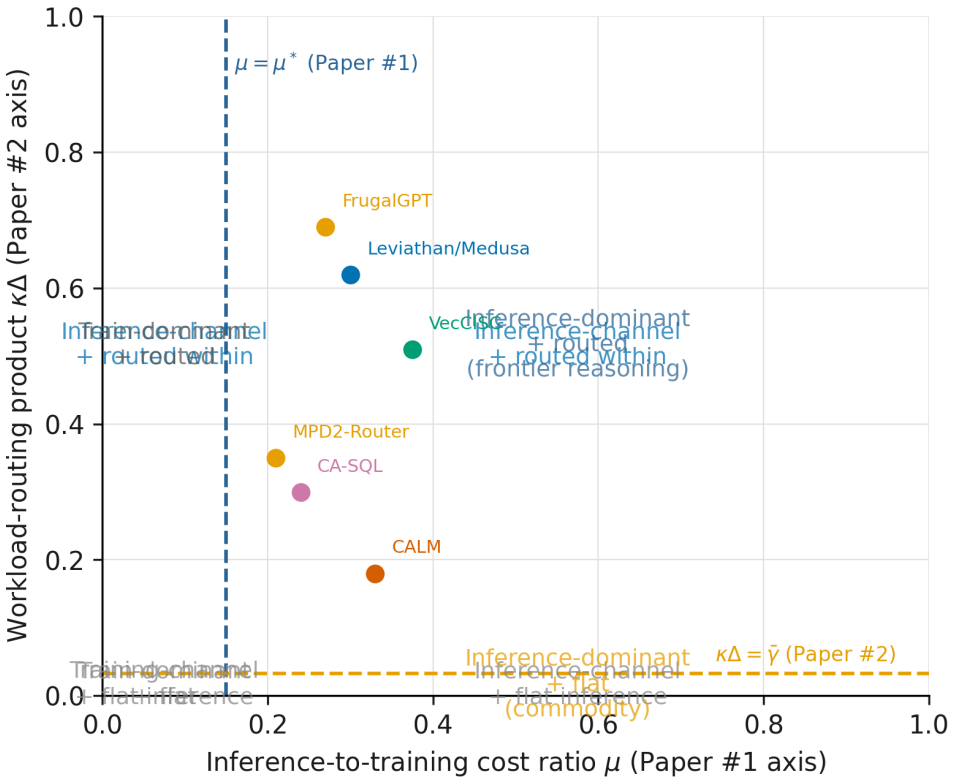

Figure 3. Composition of the Paper #1 channel-allocation threshold (vertical line at $\mu = \mu^*$: inference-to-training cost ratio vs. the Paper #1 switching threshold) and the Paper #2 workload-routing threshold (horizontal line at $\kappa\Delta = \bar{\gamma}$). The four quadrants correspond to the four combinations of training-dominant or inference-dominant channel choice and flat or routed within-inference allocation. Frontier reasoning tiers sit in the upper-right quadrant: inference-dominant and heterogeneous enough to route. Commodity tiers sit in the lower-right: inference-dominant but flat in compute allocation. The six Paper #2 calibration points are projected onto the $\kappa\Delta$ axis. Reproducibility: `paper-2/figures/fig3_composition.py`.

multiplicatively. The combined production cost satisfies

$$C_{\text{total}} \;=\; C(\bar{c}^*, \bar{d}) \cdot \left[1 - \Pi_1\right]^{\mathbf{1}[\text{ch = inference}]} \cdot \left[1 - \Pi_2\right]^{\mathbf{1}[\kappa\Delta > \gamma]}, \tag{10}$$

where Π_1 is the Paper #1 routing premium (inference channel relative to training channel) and $\Pi_2 = \kappa\Delta - \gamma$ is the Paper #2 routing premium. The two indicators are independent: the inference-channel decision (Paper #1) gates on query difficulty and training-cost structure; the workload-routing decision (Paper #2) gates on workload heterogeneity and classifier overhead. Figure 3 plots both thresholds on the same axes, with the four quadrants labeled by the implied allocation regime.

The composition has a serving implication. A provider that has crossed Paper #1's threshold (i.e., it is already cost-optimal to invest in inference-time scaling) will also want to cross Paper #2's threshold if the workload is heterogeneous enough. The two conditions can both be satisfied at the same operating point, and in the production workloads we examine ($\Delta \gg \gamma/\kappa$, $\eta_\alpha^\rho - 1 \gg 1/\mu$) they are both satisfied simultaneously. The combined cost reduction is multiplicative and larger than either reduction alone.

6. Conclusion

The routing premium $\kappa\Delta > \gamma$ is positive at the margin around the unconditional optimum when the classifier calibration and the workload heterogeneity together exceed the classifier overhead. We derive the condition from the Cost-correct framework, show it nests the five major published instances of difficulty-conditional compute as corollaries, and calibrate it against six operating points spanning all five classes. Every calibrated operating point sits on the positive side. CALM, as the early-exit representative, sits closest to the threshold: its per-layer Δ is bounded by depth-curvature, making it the sensitivity case that constrains the useful operating range of exit confidence calibration.

The derivation has two open edges. First, κ at production scale is not directly observable from public APIs: the explained-variance calibration of a provider's internal difficulty classifier is not disclosed in any of the six papers we calibrate, and we infer it from proxy observables. Second, the second-order local result is sufficient when the classifier policy stays close to the unconditional optimum, but cascade and deferral systems that make large discrete jumps in compute can violate the local approximation; a global routing-premium result (incorporating all orders of the Taylor expansion) remains open.

We invite serving providers to disclose routing-accuracy distributions alongside cost-reduction reports. A disclosed κ on a representative workload sample would let independent researchers verify or falsify the threshold directly, rather than relying on the elasticity reading from proxy observables. That disclosure would also distinguish the source of cost reductions in deployed tier-menu systems: whether the gains come from calibration (κ close to 1), from workload heterogeneity (Δ large), or from a fortuitous combination of both.

7. Cite this article

```
@article{bhardwaj2026routingpremium,
  author = {Manu Bhardwaj},
  title  = {The Routing Premium.
            An Economic Threshold for
            Difficulty-Conditional Inference Compute},
  journal = {arXiv preprint},
  year   = {2026},
  url    = {https://ifitsmanu.com/papers/
              the-routing-premium},
}
```

References

[1] Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav Gulavani, Alexey Tumanov, and Ramachandran Ramjee. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve. *arXiv preprint arXiv:2403.02310*, 2024. URL https://arxiv.org/abs/2403.02310.

[2] Manu Bhardwaj. The α asymmetry. why verifiers can be smaller than generators. Field Notes #3, ifitsmanu.com, May 2026. URL https://ifitsmanu.com/papers/the-alpha-asymmetry.

[3] Manu Bhardwaj. The cost of being right. verification economics in 2026. Field Notes #2, ifitsmanu.com, May 2026. URL https://ifitsmanu.com/papers/the-cost-of-being-right.

[4] Manu Bhardwaj. The inference-time compute frontier. a cost-correct threshold for training versus test-time allocation. Working paper, ifitsmanu.com, May 2026. URL https://ifitsmanu.com/papers/the-inference-time-compute-frontier. Paper #1 of the Inference Economics wedge.

[5] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024. URL https://arxiv.org/abs/2401.10774.

[6] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023. URL https://arxiv.org/abs/2302.01318.

[7] Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023. URL https://arxiv.org/abs/2305.05176.

[8] Umutcan Erol, Jad El, Mirac Suzgun, Mert Yuksekgonul, and James Zou. The cost of being right: Evaluating language models by the cost-of-pass. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vC9S20zsgN.

[9] Joon Ha Kim, Geon-Woo Kim, Anoop Rachakonda, and Daehyeok Kim. Dooly: Configuration-agnostic, redundancy-aware profiling for LLM inference simulation. *arXiv preprint arXiv:2605.07985*, 2026. URL https://arxiv.org/abs/2605.07985.

[10] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2211.17192.

[11] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024. URL https://arxiv.org/abs/2401.15077.

[12] Christian Lysenstøen. SLO-Guard: Crash-aware, budget-consistent autotuning for SLO-constrained LLM serving. *arXiv preprint arXiv:2604.17627*, 2026. URL https://arxiv.org/abs/2604.17627.

[13] Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Inigo Goiri, Saeed Maleki, and Ricardo Bianchini. Splitwise: Efficient generative LLM inference using phase splitting. *arXiv preprint arXiv:2311.18677*, 2024. URL https://arxiv.org/abs/2311.18677.

[14] James Petullo and Nianwen Xue. CA-SQL: Complexity-aware inference time reasoning for text-to-SQL via exploration and compute budget allocation. *arXiv preprint arXiv:2605.08057*, 2026. URL https://arxiv.org/abs/2605.08057.

[15] James Petullo, Sonny George, Dylan Cashman, and Nianwen Xue. VecCISC: Improving confidence-informed self-consistency with reasoning trace clustering and candidate answer selection. *arXiv preprint arXiv:2605.08070*, 2026. URL https://arxiv.org/abs/2605.08070.

[16] Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *arXiv preprint arXiv:2207.07061*, 2022. URL https://arxiv.org/abs/2207.07061. Also in NeurIPS

2022.

[17] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314.

[18] Wenxin Zhan. MPD^2-router: Mask-aware multi-expert prior-regularized dual-head deferral router in glaucoma screening and diagnosis. *arXiv preprint arXiv:2605.08024*, 2026. URL https://arxiv.org/abs/2605.08024.

Set in Computer Modern 11/15. Typeset with LaTeX. ifitsmanu.com.