EVALUATING ESCALATION SIGNALS FOR LLM ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cascading a small language model into a large one requires a signal that predicts, before the large model is called, whether the call is needed. We study semantic entropy over k sampled answers as that signal, and evaluate it on three tasks of increasing openness: templated arithmetic, GSM8K reasoning, and text-to-SQL.

On GSM8K with a Gemma-3 1B/12B pair, semantic entropy predicts the small model's own failures at AUROC 0.871, beats a difficulty-calibrated baseline by $+0.313$, and converts into routing gains of up to $+0.093$ accuracy over random escalation at matched cost. On synthetically generated chained arithmetic it adds nothing over a regular expression on the question text: a single-family benchmark makes difficulty a free surface feature of the question, so sampling reveals no more than the question already gives away. On text-to-SQL (Gemma-3 4B/12B, n=1200), semantic entropy scores only 0.545 AUROC pooled, below a free difficulty baseline (0.850) – but it is undefined, not weak, on the 77% of items where the k samples agree; on the remaining 23% it scores 0.676. Splitting on this free-to-compute distinction and calibrating each half before recombining reaches a held-out AUROC of 0.907, our strongest result. A retrieval alternative – caching large-model outcomes and querying by embedding similarity – trades signal for cost: it appears to predict large-model success on arithmetic (0.908), but a probe shows this is again difficulty lookup (ρ=0.98), and it collapses to chance on GSM8K (0.518), where the same probe returns ρ=0.23. It costs $0.57\times$ always-large against $0.82\times$ for live sampling at its tuned budget.

Framed as prediction rather than diagnosis, semantic entropy earns its keep only where the difficulty of a query cannot be read off the question itself. We report the diagnostics that produced these outcomes – a difficulty baseline, an oracle-headroom check, per-generation cost accounting, a retrieval-difficulty probe, and a regime split – against a measured AUROC noise floor of ≈ 0.04 at these sample sizes.

1 INTRODUCTION

Serving every query with the largest available language model is wasteful: many queries are easy enough for a small model (SLM) to answer correctly, and only a minority need the extra capacity of a large model (LLM) or a larger SLM. A router that can cheaply predict, before querying the large model, whether a given query needs it would let a system pay the large model's cost only when it is worth paying (Figure 1).

Semantic entropy (Farquhar et al., 2024) is a natural candidate signal. Originally proposed for hallucination detection, it estimates a model's uncertainty about the *meaning* of its own answer: draw k stochastic samples for a query, cluster them by bidirectional NLI entailment, and compute the Shannon entropy of the resulting cluster-probability distribution. High semantic entropy indicates that the model's samples disagree on meaning, not just wording.

This paper asks: can semantic entropy, computed from the small model's own samples, predict whether escalating a query to a large model would improve the answer? We build a full pipeline to test this across five benchmarks and two model families, and report both positive and negative results, several real implementation bugs encountered along the way, and a conceptual clarification of what semantic entropy can and cannot be expected to predict. Our contribution is not the signal

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(a) Scored as a routing decision: *which model should serve this query?* The router must anticipate the large model's behavior – unavailable when the decision is made. **Target 1: AUROC 0.693.**

(b) Scored as a failsafe: *is my own answer wrong?* The check reads only the small model's own samples. **Target 2: AUROC 0.865.**

Figure 1: Two questions a cascade can ask of the same signal. Both panels show the same semantic-entropy run at k=5 on GSM8K; framing alone moves measured performance by ≈ 0.17. A cascade read as a cost-minimizing router is assessed on the harder of the two questions.

but the evaluation methodology: we identify six controls that escalation studies do not currently report, each of which reversed one of our own conclusions when applied.

2 Related Work

Cascades and routing. Cascading was popularized by FrugalGPT (Chen et al., 2023), which learns confidence thresholds over a portfolio of APIs. AutoMix (Aggarwal et al., 2024) wraps few-shot self-verification in a POMDP-based router. RouteLLM (Ong et al., 2025) and Hybrid LLM (Ding et al., 2024) both perform pre-generation routing, selecting a model before any inference – a distinction that matters for our cost analysis, since a pre-generation router never pays the small model, whereas a cascade pays it on every query.

Uncertainty signals for deferral. Token-level confidence is the established deferral signal for generative cascades; Gupta et al. (2024) formalize the Chow-style variants our `avg_neg_logprob` instantiates. Semantic uncertainty (Kuhn et al., 2023) and its refinement into semantic entropy (Farquhar et al., 2024) were introduced for hallucination detection rather than routing.

Agreement-based deferral. The closest prior work is Soiffer et al. (2025), who use semantic agreement among an ensemble of distinct small models as a training-free deferral signal. Kolawole et al. (2025) use ensemble agreement for cascade deferral and Yue et al. (2024) use answer consistency for numerical reasoning. Our setting differs in construction – k stochastic samples from a *single* model, clustered by entailment – but the underlying intuition is shared; we regard the effectiveness of this family as established by that work rather than as our contribution.

What this paper adds. To our knowledge, no prior cascade study reports a difficulty baseline, an oracle-headroom precondition, the difficulty-calibrated variant needed to evaluate retrieval-based routers fairly, or the observation that on lexically regular benchmarks an embedding space can encode difficulty so completely ($\rho = 0.98$) that nearest-neighbor retrieval is difficulty lookup. Selective prediction (Geifman & El-Yaniv, 2017) supplies the abstention framework these controls operate within.

3 Method

Semantic entropy. Given query q and k stochastic samples $y_1, \ldots, y_k$ at temperature > 0, we cluster via bidirectional entailment using an NLI model (`microsoft/deberta-large-mnli`): y_i, y_j share a cluster iff $P(\text{entail} \mid y_i, y_j) > \tau$ and $P(\text{entail} \mid y_j, y_i) > \tau$, $\tau = 0.5$. Writing $p(C_c) = |C_c|/k$,

$$H = -\sum_c p(C_c) \log p(C_c), \qquad H_{\text{norm}} = H/\log k \in [0, 1]. \tag{1}$$

Clustering need not use NLI: for a tool-calling agent (e.g. text-to-SQL, Section 4.6), equivalence is decided by *execution* instead – two generations mean the same thing iff they return the same rows. This eliminates τ as a parameter and removes stylistic variation as a confound entirely.

Token-level confidence. We also record each sample's average per-token log-probability, negated: `avg_neg_logprob`, for the representative sample from the majority cluster – a different notion of uncertainty that can catch a model that is consistently wrong (zero semantic entropy) but assigns low probability to its own output.

Three targets. For each query we compute `correct_small` and `correct_large`. Three binary targets follow:

$$\text{T1: needs_escalation} = (\neg\texttt{correct_small}) \wedge \texttt{correct_large}, \tag{2}$$

$$\text{T2: untrustworthy} = \neg\texttt{correct_small}, \tag{3}$$

$$\text{T3: large_success} = \texttt{correct_large}. \tag{4}$$

T1 is the operationally relevant question, but depends on the large model's behavior, which a small-model-only signal never directly observes – it may still carry *correlational* information via shared query difficulty. T2 is the target a per-model signal can fairly be judged against. T3 applies only to signals that observe the large model (Section 4.5).

Cost model. Normalizing the k-sample pass to 1 unit, $\text{cost}(r) = 1 + \rho r$ for escalation rate r and cost ratio ρ. We compare against always-small, always-large, and random-at-matched-cost. This normalization hides a binding constraint (Section 3).

The sampling overhead this cost model hides. Charging per generation, with a large-model call costing $\gamma\times$ a small-model one,

$$\text{cost}_{\text{router}} = k + r\gamma, \qquad \text{cost}_{\text{always-large}} = \gamma, \qquad \text{routing wins only if } \gamma > \frac{k}{1-r}. \tag{5}$$

At k=10 this requires $\gamma > 11.1$ even at r=0.1. Viability depends on the model pair, not signal quality alone, and is easy to miss under the normalized model.

The difficulty baseline: a necessary control. An escalation signal is only interesting if it beats what the question already reveals for free. *Raw difficulty* scores each item by the benchmark's own difficulty value directly, which assumes the target is monotonic in difficulty. *Difficulty-calibrated* instead estimates $P(\text{label} \mid \text{difficulty})$ via leave-one-out neighbors, recovering non-monotonic structure at the cost of estimation noise. Neither dominates in general; we report both and take the larger as the bar a signal must clear.

4 Results

4.1 Two benchmarks, one signal, opposite verdicts

Synthetic arithmetic, the cautionary case. On a chained-arithmetic ladder, semantic entropy separated the fair target (T2) at AUROC 0.830 and 0.867 across two runs – accepted initially as a positive result. Scoring the benchmark's own difficulty directly (a regex over the question, at zero cost) gives 0.839 and 0.830: a tie, sign unstable, both gaps inside the 0.04 noise floor (Section 4.4). Oracle headroom over always-large is only +0.021. *We claim something narrower than "a regex is the better router."* Writing that regex presupposes knowing the queries are chained arithmetic; a system facing an open workload would need to identify the task family first, which a task-agnostic signal never has to do. The finding is that a single-family benchmark confounds uncertainty with difficulty by construction.

GSM8K, where the signal survives. A Gemma-3 1B/12B pair, n=250 GSM8K items, k=10. Always-small 0.372, always-large 0.484, oracle 0.572 (+0.088 headroom, 4× synthetic).

Table 1: GSM8K, Gemma-3 1B/12B, n=250, k=10. AUROC with 95% bootstrap intervals. The difficulty baseline uses GSM8K's own difficulty annotation (calculation lines in the reference solution – a skyline, since it presupposes the answer).

Signal	T1: needs_escalation	T2 (fair): ¬correct_small	T3: ¬correct_large
Semantic entropy	0.734 [0.668, 0.799]	**0.871** [0.820, 0.915]	0.639 [0.570, 0.709]
Token neg-logprob	0.648 [0.562, 0.731]	0.647 [0.577, 0.711]	0.490 [0.419, 0.561]
Combined	**0.749** [0.685, 0.813]	0.852 [0.797, 0.900]	0.608 [0.536, 0.678]
Difficulty (raw)	0.522 [0.444, 0.603]	0.649 [0.584, 0.712]	0.633 [0.564, 0.702]
Difficulty-calibrated	0.499 [0.416, 0.583]	0.558 [0.486, 0.629]	0.535 [0.460, 0.610]

Entropy beats raw difficulty by +0.212 (T1) and +0.222 (T2), and calibrated difficulty by +0.235/+0.313, with disjoint CIs on T2. Where the synthetic gap was inside the noise floor, these

margins exceed it fivefold. The difficulty baseline here is a *skyline*, not a deployable competitor – it presupposes the reference solution – yet entropy, observing only the small model's own samples, beats it. Routing converts this into real gains (Table 2): accuracy exceeds always-large for escalation rates 0.37–0.89, peaking at $+0.093$ over random-at-matched-cost. A second, disjoint n=150 sample replicates T2 to within 0.008 AUROC.

Table 2: GSM8K Pareto curve, routing on semantic entropy. Always-large accuracy $= 0.484$.

Esc. rate	Accuracy	Random	Margin	vs. large	Harmful	Wasted
0.00	0.372	0.372	−0.000	−0.112	0.000	0.000
0.37	0.500	0.413	+0.087	+0.016	0.022	0.087
0.56	**0.528**	0.435	**+0.093**	+0.044	0.043	0.135
0.89	0.496	0.473	+0.023	+0.012	0.085	0.300
oracle	0.572	–	–	+0.088	0.000	0.000

4.2 The sample budget: ablating k

k=10 fails the break-even condition of Eq. 5 (costs 1.38× always-large). Re-running at $k \in \{3, 5, 10\}$: T2 AUROC is $0.791/0.865/0.862$ – k=5 and k=10 agree to 0.003, an order of magnitude inside the noise floor, while k=3 costs 0.074. Always-small accuracy falls with k too $(0.312/0.364/0.392)$, since the pipeline reports the majority-cluster answer, so k doubles as a self-consistency ensemble. **At k=5, routing strictly dominates always-large**: cost 9.80–11.6 units against always-large's 12.0, at equal or better accuracy, versus k=10's 16.6 units (never competitive). *The barrier we reported as a property of the method was a hyperparameter we had never varied.* At k=3, token confidence overtakes entropy on T1 (0.658 vs. 0.630) – entropy degrades with fewer samples, confidence does not, and their fusion's advantage widens as k shrinks.

4.3 Extended replication tables

GSM8K on a second, disjoint sample. Table 3 gives full intervals for the n=150 replication referenced in Section 4.1. On T2, entropy's interval is disjoint from both difficulty baselines in *both* samples, so the margin is not a sampling artifact. On T1 that separation holds only at n=250; at n=150 entropy's interval ($[0.549, 0.743]$) overlaps raw difficulty's ($[0.414, 0.607]$), so the T1 margin should be read as directional rather than established. Token confidence is not universally anti-informative here: it reaches 0.648/0.647 on T1/T2 (weak, but above chance), the reverse of what synthetic arithmetic showed.

Table 3: GSM8K on two disjoint samples, same Gemma-3 1B/12B pair. AUROC with 95% percentile bootstrap intervals (2000 resamples). Oracle headroom is a point estimate, not bootstrapped.

	n=250	n=150
T2: semantic entropy	0.871 [0.820, 0.915]	0.863 [0.790, 0.926]
T2: difficulty (raw)	0.649 [0.584, 0.712]	0.621 [0.533, 0.708]
T2: difficulty (calibrated)	0.558 [0.486, 0.629]	0.606 [0.512, 0.705]
T1: semantic entropy	0.734 [0.668, 0.799]	0.648 [0.549, 0.743]
T1: difficulty (raw)	0.522 [0.444, 0.603]	0.508 [0.414, 0.607]
T1: difficulty (calibrated)	0.499 [0.416, 0.583]	0.497 [0.398, 0.602]
Oracle headroom over always-large	+0.088	+0.093

Synthetic arithmetic, properly powered. Table 4 gives the 96-item re-analysis behind the null result of Section 4.4. The two datasets share their item set and model pair, differing only in resampling: the greedily-decoded large model returns identical labels in both (always-large 0.656), while the small model's k samples flip two of 96 items (always-small 0.385/0.365). This isolates run-to-run variance from item-sampling variance, and is the source of the 0.04 noise floor used throughout this paper. Token confidence is anti-informative here in all four measurements (0.413–0.433), the opposite of its sign on GSM8K (Table 3) – a reversal we take as evidence that a signal's sign is a property of the benchmark, not a fixed property of the signal.

Table 4: The same 96-item synthetic set profiled twice, 95% bootstrap CIs. Difficulty is recoverable from the question by regex, so the baselines are free competitors, not skylines.

Target	Signal	Dataset A	Dataset B
T1: needs_escalation	Semantic entropy	0.613 [0.502, 0.726]	0.666 [0.554, 0.774]
	Token neg-logprob	0.413 [0.289, 0.542]	0.381 [0.270, 0.502]
	Difficulty (raw)	0.563	0.559
T2 (fair): ¬correct_small	Semantic entropy	**0.830** [0.742, 0.904]	**0.867** [0.787, 0.934]
	Token neg-logprob	0.431 [0.310, 0.549]	0.433 [0.311, 0.556]
	Combined	0.752 [0.642, 0.854]	0.759 [0.651, 0.858]
	Difficulty (raw)	0.839	0.830
T3: ¬correct_large	Semantic entropy	0.722 [0.601, 0.838]	0.714 [0.594, 0.827]
	Difficulty (raw)	0.791	0.791

The full k-ablation and cost frontier. Table 5 gives all three targets across $k \in \{3, 5, 10\}$ underlying the summary in Section 4.2; Target 3's difficulty baseline is identical across all three rows (0.633), an internal check that the item set did not change with k. Table 6 gives the cost-accuracy frontier in full: at k=10 no configuration dominates always-large, while three configurations at $k \leq 5$ do.

Table 5: Ablating the sample budget on GSM8K (Gemma-3 1B/12B, n=250), AUROC with 95% bootstrap intervals.

Target	Signal	k=3	k=5	k=10
T1: needs_escalation	Semantic entropy	0.630 [0.562,0.698]	**0.693** [0.624,0.758]	0.682 [0.612,0.747]
	Token neg-logprob	0.658 [0.583,0.731]	0.632 [0.548,0.711]	0.628 [0.550,0.707]
	Combined	0.671 [0.599,0.745]	0.720 [0.652,0.788]	0.699 [0.632,0.766]
	Difficulty (raw)	0.517	0.522	0.560
T2 (fair)	Semantic entropy	0.791 [0.730,0.850]	**0.865** [0.814,0.909]	0.862 [0.812,0.906]
	Token neg-logprob	0.651 [0.580,0.721]	0.645 [0.573,0.711]	0.613 [0.540,0.682]
	Combined	0.804 [0.740,0.865]	0.852 [0.800,0.897]	0.819 [0.761,0.872]
	Difficulty (raw)	0.665	0.663	0.677
T3	Semantic entropy	0.627 [0.558,0.688]	0.644 [0.575,0.711]	0.684 [0.616,0.749]
	Difficulty (raw)	0.633	0.633	0.633
Always-small accuracy		0.312	0.364	0.392
Oracle headroom over always-large		+0.068	+0.104	+0.112

Table 6: Cost–accuracy frontier across k on GSM8K, γ=12. Cost is $k + r\gamma$ per query in small-model-generation units; always-large costs 12.0 at accuracy 0.484.

k	Signal	r	Accuracy	Cost	Dominates always-large?
3	combined	0.70	0.500	11.40	yes
5	entropy	0.55	0.520	11.60	yes
5	combined	0.40	0.508	9.80	yes
5	combined	0.50	0.520	11.00	yes
10	entropy	0.55	0.512	16.60	no
10	combined	0.60	0.516	17.20	no

4.4 Synthetic arithmetic: the null result, and why it is one

Repeating the synthetic run on two independently profiled 96-item sets with 95% CIs: entropy separates T2 at $0.830/0.867$, difficulty at $0.839/0.830$ – statistically indistinguishable, gaps inside a measured 0.04 AUROC noise floor. Entropy's rank correlation with difficulty is $\rho = 0.657$; its Somers' D against the T2 label is 0.660 – agreement to three decimal places, indicating its discriminative power *is* difficulty tracking. This is a null specific to single-family benchmarks, where difficulty is a free surface feature by construction, not a claim that entropy fails in general

(Section 4.1 shows the opposite on GSM8K). PopQA produced a different failure: 72% both-wrong items (a parametric knowledge gap, not an uncertainty problem), leaving zero oracle headroom for any signal.

4.5 KNOWLEDGE-BASE RETRIEVAL: A CHEAPER, WEAKER SIGNAL

A retrieval-based router caching large-model outcomes appeared to predict T3 (`correct_large`) at AUROC 0.908 on synthetic arithmetic. A leave-one-out probe – predicting an item's difficulty from its neighbors' difficulty, using the same vote the KB uses for outcomes – returned $\rho = 0.986$: the embedding *was* difficulty lookup. On GSM8K, the same probe returns $\rho = 0.231$ and T3 collapses to 0.518 (chance) – *predicted before that benchmark was run*. Retrieval remains the cheaper policy: $0.57\times$ always-large (one embedding) versus live sampling's $0.82\times$ at its tuned k=5 budget, so the two trade signal quality against price rather than one dominating.

4.6 TEXT-TO-SQL: RECOVERING A SIGNAL THAT POOLING HAD HIDDEN

Semantic entropy generalizes beyond NLI clustering: for a tool-calling agent, two generations mean the same thing iff they return the same rows after execution. We evaluate a Gemma-3 4B/12B pair on a hardened 7-table schema (n=1200, six query families), at a schema-disclosure level (table and column names, no types or keys) calibrated to avoid two failure modes at the extremes: full disclosure let the 12B model saturate a simpler two-table schema at accuracy 1.00 (no oracle headroom), while names-only disclosure put both models near floor ($56/60$ generations failing to execute for a 1B/4B pair). Entropy scores 0.545 AUROC on T2 – *below* raw difficulty (0.850) – and appears to fail outright.

The mechanism. A concrete failure: "How many gold-tier assets at Dallas are under any contract?" The correct query is an existence check (`asset_id IN (SELECT ...)`); a plausible alternative joins directly to `contracts`, inflating the count whenever an asset holds more than one contract row. A model that has *learned* to reach for this join does so on every sample – the error recurs identically across all k generations, so entropy reads exactly 0 despite the answer being wrong every time.

Table 7: Text-to-SQL, T2, split by whether the k samples disagreed at all (executable items only, n=954). "–" means entropy is undefined, not zero.

Regime	Share	Wrong rate	Entropy AUROC	Neg-logprob AUROC
Diverse (num_clusters>1)	22.7%	0.903	**0.676**	0.520
Uniform (num_clusters=1)	77.3%	0.773	–	**0.807**

Entropy is informative on its intended regime (0.676) and undefined – not weak – elsewhere; the pooled 0.545 is an artifact of averaging a real signal against a population where it mathematically cannot apply. A *naive* gate (entropy when diverse, confidence otherwise, rank-normalized per regime) *underperforms* confidence alone once pooled (0.793 vs. 0.808): matched ranks within a regime do not imply matched probabilities across regimes. Calibrating each regime to $P(\text{wrong} \mid \text{score})$ via leave-one-out before pooling recovers it decisively.

Table 8: Gated, calibrated signal vs. strongest prior baseline. 2-fold held-out calibrates each item from the opposite fold only.

Target	Best prior baseline	Leave-one-out	2-fold held-out
T1: needs_escalation	0.670	0.833	0.799
T2 (fair)	0.861	0.973	**0.907**
T3	0.714	0.893	0.850

The held-out estimate is 0.03–0.07 lower than leave-one-out (mild shared-pool leakage) but survives decisively rather than collapsing. This signal is not zero-label like entropy or confidence – it needs a reference pool, like the difficulty-calibrated baseline – but the mechanism it establishes is general: a

cheap, already-computed feature (`num_clusters`) identifies which of two complementary signals to trust, and calibration, not gating alone, recovers the value pooling discards.

4.7 PopQA: escalation is the wrong intervention

PopQA proved unusable for this study, for a reason worth separating from ordinary experimental difficulty. An extraction/grading mismatch was first suppressing measured accuracy by roughly half: our pipeline reduced each generation with a `last_sentence` extractor before grading with a substring matcher, but the matcher searches for a short answer embedded in prose, so extracting the final clause first discards the text it was built to search. Grading the raw generation instead raised Gemma-3 12B from 0.12 to 0.28 and 1B from 0.04 to 0.12 on matched pre-flight samples – a silent bug that presents as uniformly weak models rather than as an error.

Correcting it does not rescue the benchmark. The corrected numbers still leave 72% of items wrong for both models, with zero pre-flight items on which the small model succeeded where the large one failed. The dominant failure mode on long-tail PopQA is absent parametric knowledge in both models, not small-model uncertainty, exactly as the dataset's design rationale anticipates (Mallen et al., 2023). Escalation cannot repair an answer neither model knows; we regard PopQA as misspecified for escalation research rather than as a negative result about semantic entropy, and take the general lesson: a large small/large capability gap does not imply escalation has anything to gain, since what matters is whether the large model succeeds specifically where the small one fails — oracle headroom, not the raw gap.

4.8 A learned alternative: the confidence-gated router

The signals above are all fixed statistics. We also tested a learned one: a probe trained directly on `needs_escalation`, reading the small model's mid-layer hidden states (layers 8–16, ensembled, ℓ_2-regularized), with a proper train/calibration/test split rather than the single-pass evaluation used elsewhere in this paper. A probe can exceed a fixed statistic here because it reads the representation samples were generated from, which encodes properties of the query — topic, form, apparent difficulty — that also bear on whether the large model will succeed; the small model still never observes the large one, but the probe exploits the indirect route more effectively than a summary statistic can.

On a 72-item synthetic-arithmetic benchmark (Gemma-3 270M/4B), an eight-iteration search raised AUROC against `needs_escalation` from 0.733 (semantic entropy) to 0.895, and routed accuracy from 0.577 to 0.615 while escalating only 30% of queries — roughly 80% of the gain an oracle could achieve. This is compatible with, not contradictory to, the null result of Section 4.4: a fixed, parameter-free signal cannot reliably predict `needs_escalation` at scale, but a signal with a learned component, trained directly against that label on internal representations, recovers separation the fixed signals cannot reach. This independent pipeline also found that fusing neg-logprob into the combined score hurt performance, corroborating our main finding that fusion should be validated per dataset rather than assumed to help.

5 Discussion

Semantic entropy is a usable escalation signal, but establishing that took four benchmarks and the repeated discovery that our own positive results were artifacts. The generalisable content of this paper is the set of controls that separated the two.

Report the target explicitly. Target 1 (*does escalation help*) and Target 2 (*is the small model wrong*) differ by more than 0.2 AUROC on identical data. A signal computed from the small model's samples has only indirect access to the large model's behaviour, mediated by shared query difficulty — substantial, as Target 3 shows (0.72 on synthetic), but not the same thing as observing the large model. Papers that report one target without naming it are not comparable to each other.

Report a difficulty baseline. This is the control we most regret omitting. On synthetic arithmetic a regex over the question matched semantic entropy exactly, converting our strongest result into a null one; on GSM8K entropy beats the same baseline by $+0.222$ and the result stands. The two

cases are indistinguishable without the baseline. Any signal correlated with difficulty will separate a difficulty-driven target, so the question is never "does the signal work" but "does it beat what the question text already tells you."

Check headroom before spending compute. A benchmark can exhibit a wide capability gap and still leave a perfect router $+0.021$ over always-large, because the relevant quantity is $P(\text{correct_small} \wedge \neg\text{correct_large})$ — how often the small model is right where the large model is wrong — not the accuracy difference between the models. On PopQA that quantity was zero. This is a cheap pre-flight computation and it invalidates entire experiments.

Cost the sampling overhead honestly. Normalising the k-sample pass to one unit hides that entropy routing pays k generations on every query before deciding anything. Under per-generation accounting the method saves money only when $\gamma > k/(1-r)$, which at $k = 10$ demands a large model more than $11\times$ the per-generation cost of the small one even at a 10% escalation rate. Our GSM8K configuration wins on accuracy and loses on cost — both worth reporting, and the second invisible under the normalised model.

On generalising across datasets. Three of our findings reversed between benchmarks, which is the most uncomfortable result in this paper. Token confidence was below chance on synthetic arithmetic and above it on GSM8K; fusing it degraded entropy in the first case and improved Target 1 in the second; and retrieval-based prediction of large-model success went from 0.908 to chance. In each case the single-benchmark result would have been reported as a property of the signal. We now think it should not be: a signal's behaviour here is a joint property of the signal and the benchmark's structure, and claims of the form "signal X predicts Y" should be accompanied by the controls that establish which of the two is doing the work.

The retrieval case is the one where we can say what the benchmark contributed. Its difficulty was lexically recoverable, embedding similarity therefore encoded difficulty, and neighbour voting reduced to difficulty-conditioned outcome frequency — an estimator our own difficulty-calibrated baseline computes without any embedding. That the probe forecast the collapse before we ran GSM8K is, to us, the clearest argument for reporting these controls by default.

Limitations. Our positive result rests on one dataset and one model pair. It does replicate across two disjoint samples of that dataset ($n = 250$ and $n = 150$, agreeing to 0.008 AUROC on Target 2), but two samples from GSM8K are not two datasets, and nothing here establishes that the result transfers to another task family or another model pair. Sampling temperature was fixed at 0.7 throughout and never ablated, despite directly setting the diversity that semantic entropy measures. We ablated the sample budget k (Section 4.2) but not the NLI entailment threshold τ, which was fixed at 0.5 throughout. The knowledge-base result of Section 4.5 additionally depends on the profiling and evaluation sets being disjoint, which we verify by question-string comparison but which is weaker than sampling them independently. Along the way we caught and fixed a grading bug that discarded correct answers before matching (halving measured accuracy for both models on a QA benchmark), an embedding-model mismatch between profiling and inference that inverted a knowledge-base result, a stylistic-elaboration confound in raw-generation clustering that inflated entropy to near-maximum despite near-unanimous agreement on content, an NLI question-context wrapper out-of-distribution for a general-purpose entailment model, a `bfloat16` dtype mismatch that crashed an NLI model's attention kernel, and a prompt that suppressed a large model's real advantage by forcing single-shot mental arithmetic with no scratchpad. Finally, the failure modes cataloged here are the ones we happened to detect; the reason we present them as a methodological contribution is precisely that each was, for some period, invisible in results that looked reasonable.

6 Conclusion

Semantic entropy can serve as an escalation signal, but only once its evaluation is designed to distinguish where it is informative from where it is silent by construction. On GSM8K it separates the small model's own failures at AUROC 0.871 and converts into routing gains of $+0.093$ accuracy at matched cost, replicating across disjoint samples. Reaching this required discarding two results that looked equally convincing: a synthetic-arithmetic AUROC matched exactly by a free regex, and a

retrieval capability that collapsed to chance exactly where a prospective probe predicted it would. The text-to-SQL study extends this lesson: an apparent failure (0.545) was entropy being undefined on 77% of items, not weak, and calibrated regime-gating recovered a held-out AUROC of 0.907 – the strongest result in this paper, born from a result that first looked like the weakest.

We identify six controls, each of which reversed one of our own conclusions: target choice, difficulty baselines (raw and calibrated), oracle headroom, per-generation cost accounting, a retrieval-difficulty probe, and regime-conditional decomposition of a pooled score. Four require no model calls at all. Two findings we regard as portable beyond this signal: cost and signal quality are separable barriers that no single method here cleared on every task, and a signal's measured behavior is inseparable from the benchmark's structure and how its evaluation is pooled. We do not claim these controls are exhaustive – they are the ones our own errors happened to expose – but they are cheap, and at least one of them, applied to a result that first looked like failure, produced the strongest finding in this paper rather than merely a caveat on it.

AI USE STATEMENT

In preparing this submission from our longer report, we used a generative AI assistant (Claude) for tasks with *recommended* disclosure under the ICLR 2027 AI Policy: editing for readability and length, formatting references and the bibliography, and drafting this statement itself. All AI-assisted edits were reviewed by the authors against the source report before inclusion. No AI was used in data generation (beyond what's declared), development of the experimental methodology, formulation of hypotheses, implementation of code, or analysis of results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The signal definitions, cost model, and evaluation targets are given in full in Section 3 (Equations 1–5); the difficulty baselines (raw and calibrated) are defined in Section 3. Model pairs, sampling parameters (k, temperature, top-p/top-k, the NLI entailment threshold τ), and generation lengths are listed in Section 3. Bootstrap confidence intervals use 2000 percentile resamples throughout, as noted in each table caption.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, et al. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2023.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations (ICLR)*, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 2024.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

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

Steven Kolawole, Don Dennis, Ameet Talwalkar, and Virginia Smith. Agreement-based cascading for efficient inference. *Transactions on Machine Learning Research*, 2025.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

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

Duncan Soiffer, Steven Kolawole, and Virginia Smith. Semantic agreement enables efficient open-ended LLM cascades. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP)*, pp. 2499–2537, 2025.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *International Conference on Learning Representations (ICLR)*, 2024.