COVERAGE IS NOT UTILITY: AN EXACT CEILING ON SELECTIVE RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation degrades when the corpus does not cover the query, and the usual remedy is a gate that estimates retrieval confidence and falls back to parametric generation. Coverage is a useful intermediate variable, but it does not determine whether retrieval helps. Using HotpotQA-distractor and MuSiQue, which ship gold and distractor paragraphs for every question, we toggle evidence availability while holding question, distractors and pipeline fixed, obtaining ground truth for both *coverage* (is the evidence indexed?) and *utility* (does grounding beat answering from parameters?). The two diverge: grounding helps on only 29.4% of questions whose evidence is indexed. Because we control coverage, we can evaluate a gate that knows it exactly, and it reaches only AUC 0.698 at predicting utility on HotpotQA and 0.757 on MuSiQue. This ceiling binds every score that carries no information about utility beyond coverage, cannot be raised by recalibration, and decomposes in closed form into two rates measurable without building any gate: how often grounding helps when evidence is and is not indexed. Reranker confidence, the standard signal, detects coverage (AUC 0.734) but predicts utility at only 0.602–0.675 and carries no detectable utility signal within the covered set, so the ceiling binds it. Asking the generator whether the passages suffice does carry such signal on HotpotQA and reaches 0.639–0.724, though its advantage over the reranker is significant on one of three backbones and absent on MuSiQue; we offer it as an existence proof, not a better gate.

1 INTRODUCTION

Retrieval-augmented generation (RAG) is usually evaluated where retrieval helps. Deployed systems get no such guarantee, and when the corpus lacks the answer, retrieval does not degrade gracefully — irrelevant passages crowd the context and displace what the model already knew.

The natural remedy is selective retrieval: estimate whether retrieval succeeded and fall back when it did not. Most instantiations treat this as a confidence problem, using reranker scores or retrieval margins as proxies for "is this evidence good?". Coverage is a valid intermediate variable, but it is not sufficient to characterise whether retrieval helps. Selective retrieval therefore contains two distinct prediction problems: detecting whether evidence is available, and predicting whether grounding will improve the model's answer. We make the second one measurable and ask how far solving the first can go toward it.

We keep the two questions separate:

- **Coverage**: is the evidence needed to answer this question in the index?
- **Utility**: does grounding in retrieved text actually produce a better answer than answering from parameters?

Existing gates are built to detect coverage, and what a deployed system needs is utility. The two come apart because retrieved gold evidence buys nothing when the model already knew the answer — and that is the common case, not the exception.

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.

Design. HotpotQA-distractor gives each question ten paragraphs, exactly two of them gold; MuSiQue gives twenty with the supporting ones flagged. Indexing all of them (GOLD) or only the distractors (NOGOLD), with question, distractors and pipeline held fixed, makes accuracy at any coverage level exact rather than simulated and supplies ground truth for the quantity gates try to infer (Section 3).

Contributions. The mathematics in this paper is elementary; the contribution is a measurement design that makes the relevant quantities observable, and what they show.

1. **A controlled intervention with exact coverage and counterfactual utility.** Toggling gold evidence with everything else fixed gives coverage by construction, and answering each question with and without grounding defines utility directly as the counterfactual improvement from retrieval, on HotpotQA and MuSiQue (Section 3).
2. **An exact ceiling on coverage-based gating, and its decomposition.** A gate that knows coverage exactly reaches AUC 0.698 (HotpotQA) and 0.757 (MuSiQue) at predicting utility. By a short argument (Appendix B) this bounds every score s with $s \perp y \mid c$, and recalibration cannot raise it. Under a balanced design it reduces to a closed-form function of two conditional utility rates (Equation (2)) that a practitioner can measure without building a gate, and which explains why the ceiling moves across datasets and backbones (Section 4).
3. **A test of which signals the ceiling binds.** Reranker confidence detects coverage (AUC 0.734) but predicts utility at 0.602–0.675, and within the covered set its utility AUC is indistinguishable from chance in all four settings: it behaves as a coverage estimator and the ceiling binds it (Section 4).
4. **A signal that carries information beyond coverage.** Asking the generator whether the passages suffice carries utility signal within the covered set on all three HotpotQA backbones. Its overall advantage over the reranker separates from zero on one backbone only and does not hold on MuSiQue, so we offer it as an existence proof (Section 5).

Two supporting results are reported separately: silent hazards for local-inference evaluation (Section 6), and a standing-corpus deployment regime (Appendix C).

2 RELATED WORK

Retrieval-augmented generation. Lewis et al. (2020) introduced RAG and REALM (Guu et al., 2020) extended it to pre-training. Dense Passage Retrieval (Karpukhin et al., 2020) established bi-encoder retrieval as competitive with BM25 (Robertson & Zaragoza, 2009), and Fusion-in-Decoder (Izacard & Grave, 2021) showed that conditioning on many passages improves multi-hop accuracy. This line largely assumes retrieval is worth performing; we study when it is not.

Adaptive and selective retrieval. Self-RAG (Asai et al., 2024) trains a model to emit retrieval and critique tokens; FLARE (Jiang et al., 2023) retrieves when generation becomes low-confidence; SKR (Wang et al., 2023) elicits the model's self-knowledge to decide whether to retrieve; Adaptive-RAG (Jeong et al., 2024) routes by predicted query complexity; and REPLUG (Shi et al., 2024) tunes the retriever against a frozen model's feedback. Surveys (Gao et al., 2023) catalogue further heuristics built on retrieval scores and model confidence. Mallen et al. (2023) show that retrieval can hurt on questions about popular entities the model already knows and gate retrieval on entity popularity; our utility label makes the same observation per question. Our contribution is orthogonal to any specific method: by constructing the ground truth these methods try to infer, we measure the ceiling on those whose signal is informative only through coverage, and test which signals fall in that class.

Context sufficiency and irrelevant context. Joren et al. (2025) define sufficient context, build an autorater that classifies it, and use it for selective generation, deciding when to abstain. Our sufficiency gate asks the generator a similar question but uses the answer to route between grounded and parametric answers, and evaluates it against utility rather than sufficiency labels. The two differ exactly when the model already knows the answer: the context is then sufficient but not useful. Irrelevant passages are known to distract language models (Shi et al., 2023), and training can make

them more robust (Yoran et al., 2024); the cost of retrieving without coverage in Table 1 is an instance of this.

Reranking. Cross-encoder reranking (Nogueira & Cho, 2019) is standard in precision-oriented pipelines. Reranker scores are trained to *order* candidates within a list. We find they are respectable coverage detectors and weak utility predictors: they solve the problem they are trained for, and that problem is coverage rather than utility.

Calibration. Post-hoc calibration, notably temperature scaling (Guo et al., 2017), fixes the *scale* of a confidence score but not its *ordering*. Since AUC is invariant to monotone transformations, the ceilings we report cannot be raised by recalibration of any kind. That is why we frame the problem as target selection rather than calibration.

Self-knowledge in language models. Kadavath et al. (2022) show that language models carry usable signal about which questions they can answer. Our results connect that capability to retrieval gating: because utility depends on what the model already knows, a signal elicited from the model is at least competitive with one computed from the retriever, while being subject to none of the same structural bound.

3 Experimental Setup

Data and conditions. We sample 500 questions from the validation split of HotpotQA-distractor (Yang et al., 2018). GOLD indexes all ten paragraphs; NOGOLD indexes only the eight distractors. Answers are scored by exact match and token F1 under standard HotpotQA normalisation. Section 4 repeats the identical construction on 500 MuSiQue questions (Trivedi et al., 2022), whose twenty paragraphs per question carry an `is_supporting` flag, giving the same exact toggle over an independently built and larger distractor pool.

Pipeline. Qwen2.5-7B-Instruct (Qwen Team, 2024) (Q4_K_M GGUF) generates and ms-marco-MiniLM-L-6-v2 reranks; the top three paragraphs become context. Decoding is greedy. All inference runs locally on one consumer GPU (NVIDIA GeForce RTX 4060 Laptop, 8 GB), which is why the backbones are 7B-class quantised checkpoints. Section 5 replicates on Mistral-7B and Phi-3.5-mini.

Routing policies. NEVER always answers from parameters. ALWAYS always grounds. RERANKER gates on a softmax over the top-five reranked scores: at its shipped setting it retrieves when both the top-passage probability and its margin over the second passage clear fixed thresholds (Table 1). For AUCs and threshold sweeps we use the top-passage probability as its score; the margin alone is slightly weaker (Appendix E). SUFFICIENCY asks the generator whether the retrieved passages suffice to answer, estimating $P(\text{yes})$ by sampling its verdict. A policy is only a *routing decision* over two candidate answers, so we generate each answer once and let every policy select among them — all policies therefore see identical generations and differ only in routing.

Oracle. We report an oracle gate that knows per question whether grounding helps. It is the ceiling any gate could reach, and the correct reference point; ALWAYS is not an upper bound.

4 The Ceiling on Coverage-Based Gating

Coverage barely implies utility. Grounding improves the answer on only 29.4% of questions whose gold evidence *is* indexed, against 6.2% when it is not. Evidence availability shifts the odds but rarely determines the outcome — usually because the model already knew the answer, or because the passage cannot rescue a question it would fail anyway.

An exact bound. Because we control coverage, we can evaluate the strongest possible coverage-based gate: one that knows the ground truth. It reaches AUC 0.698 at predicting utility.

Policy	GOLD (EM)	NOGOLD (EM)
Never retrieve	17.8	17.8
Always retrieve	42.2	14.2
Reranker-confidence gate	40.6	14.8
Sufficiency gate	28.2	19.0
Oracle utility gate	47.2	24.0

Table 1: Exact match by policy and condition, n=500. The reranker gate accepts almost everything (91.2% of GOLD, 75.2% of NOGOLD) and therefore tracks ALWAYS. The sufficiency gate is the only policy that beats NEVER when evidence is absent, but at its default threshold it is too conservative to retain the benefit when evidence is present.

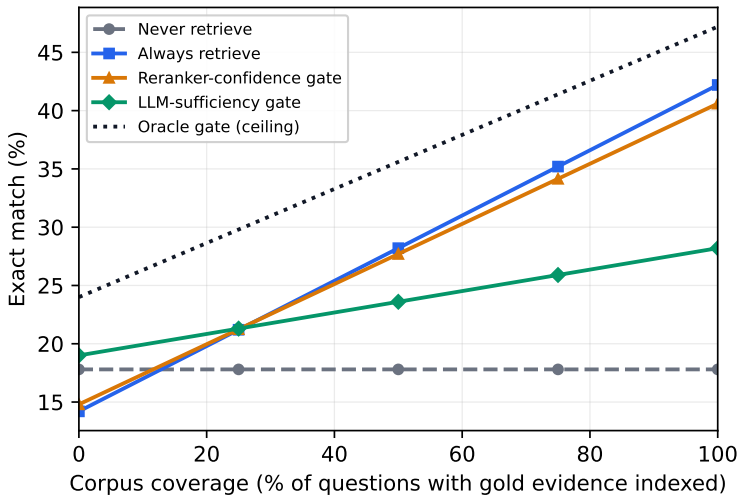


Figure 1: Accuracy against corpus coverage. Curves are exact, not simulated, because every question is measured under both conditions. A large oracle gap remains unclaimed by any deployed policy.

It is worth being precise about what this bounds. Coverage is one bit, so its AUC equals its balanced accuracy. The best gate that is a function of coverage ranks every covered question above every uncovered one, and no monotone recalibration changes an AUC. More generally, any score that is conditionally independent of utility given coverage — an estimator of coverage, however noisy, whose errors are unrelated to utility — is bounded by the same value (Appendix B gives the one-line argument). The bound does *not* apply to a score that carries information about utility within a coverage stratum. Which kind a given signal is, is an empirical question, and we test it directly below.

The bound decomposes into two rates. Reported as a single number the ceiling looks like a fact about HotpotQA. Writing $a = P(c{=}1 \mid y{=}1)$ and $b = P(c{=}1 \mid y{=}0)$ for the coverage bit c and the utility label y, midrank tie handling gives

$$\text{AUC} \;=\; a(1-b) + \tfrac{1}{2}[ab + (1-a)(1-b)] \;=\; \tfrac{1}{2} + \tfrac{a-b}{2}, \tag{1}$$

the balanced accuracy of coverage read as a predictor of utility. Let $p_1 = P(y{=}1 \mid c{=}1)$ and $p_0 = P(y{=}1 \mid c{=}0)$ be the utility rates when the evidence is and is not indexed. Our design fixes the coverage rate at one half exactly — every question contributes one GOLD and one NOGOLD row — so that rate cancels and

$$\text{AUC} \;=\; \frac{1}{2} + \frac{1}{2}\left[\frac{p_1}{p_1 + p_0} - \frac{1-p_1}{2-p_1-p_0}\right]. \tag{2}$$

Both equations are identities rather than empirical regularities: they hold exactly on any data with a balanced coverage rate, and `scripts/ceiling_closed_form.py` confirms that the implementation reproduces every measured ceiling from its own rates to within 10^{-16}. Their value is

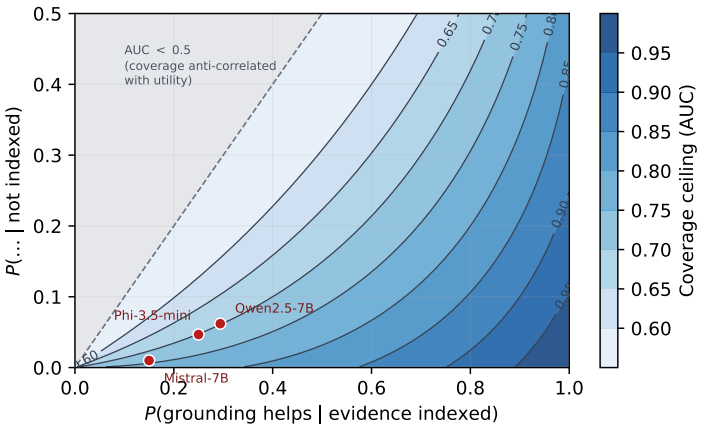


Figure 2: The coverage ceiling of Equation (2) over the two conditional utility rates, with the 3 backbones marked under greedy labels. All three sit in the low-ceiling region, and they sit there for the same reason: grounding helps on a minority of covered questions.

interpretive. The ceiling depends on the two conditional utility rates and on nothing else — not on the domain, the retriever, or the model, except through those rates.

Figure 2 plots it. The ceiling is high only where grounding almost always helps on covered questions *and* almost never helps on uncovered ones, and the measured rates are far from that corner: reaching AUC 0.90 would require grounding to help on 75% of covered questions even given a coverage oracle that never fires wrongly (p_0=0), against a highest observed value of 34.7%. It also makes the ceiling computable in advance. A reader with a different dataset need not ask whether 0.698 carries over; they can measure their own p_1 and p_0 — which requires no gate, only the ability to answer with and without retrieval — and read their own ceiling off Equation (2), so long as they hold the coverage rate at one half; away from that design the cancellation does not apply and the ceiling depends on the coverage rate as well. It also explains the spread across backbones in Table 2: Mistral-7B has the *lowest* rate of useful grounding (15.0% covered) and the *highest* ceiling (0.738), because what the bound rewards is the contrast between p_1 and p_0, not how often retrieval helps.

On a second corpus the ceiling moves where the decomposition says it should. We repeat the whole construction on MuSiQue (Trivedi et al., 2022), which ships twenty paragraphs per question with the supporting ones flagged, so the same exact toggle applies, and which was built to defeat the reasoning shortcuts HotpotQA permits. On 500 questions with Qwen2.5-7B the ceiling is 0.757 $[0.736, 0.772]$, against 0.698 $[0.670, 0.726]$ on HotpotQA: the intervals do not overlap, so the ceiling is not a constant across datasets, and we do not claim it is. What carries is the explanation Equation (2) gives for the shift. Qwen2.5-7B answers only 3.8% of MuSiQue questions from parameters, against 17.8% of HotpotQA's, so there is little for an irrelevant passage to displace and grounding almost never helps when the evidence is absent (0.4%, against 6.2%). A smaller p_0 raises the ceiling. The bound is therefore not one number to be carried between papers but a quantity each setting has, computable in advance, and higher where the model knows less. Reranker confidence does not reach it on MuSiQue either (0.600 $[0.538, 0.661]$).

Reranker confidence sits well below even this bound. Reranker top-probability is a respectable coverage detector (AUC 0.734) but reaches only 0.610 on utility. The standard signal is thus doubly limited: it does not saturate the coverage bound, and the coverage bound is itself low.

Which signals the ceiling binds. The bound applies to a score only if the score carries no information about utility once coverage is known. We test this directly by computing each signal's utility AUC among covered questions alone, where coverage is constant (`scripts/within_coverage_auc.py`; each question contributes one covered row, so a question-level bootstrap with 10,000 replicates gives the intervals). Reranker confidence is indistinguishable from chance within the covered set in every setting — 0.496 $[0.442, 0.550]$ on Qwen2.5-

		AUC at predicting utility		
Backbone	n	Reranker	Sufficiency	*Coverage ceiling*
Qwen2.5-7B	500	0.610	0.639	*0.698*
Mistral-7B	300	0.675	0.724	*0.738*
Phi-3.5-mini	300	0.602	0.706	*0.701*

Table 2: Utility-prediction AUC across backbones. Asking the generator whether the evidence suffices is above reranker confidence in point estimate on all 3 backbones, but a question-clustered bootstrap distinguishes the difference from zero on Phi-3.5-mini alone; the Phi-3.5-mini sufficiency value and its coverage ceiling are statistically indistinguishable. Point estimates only — see the text for intervals. The ceiling is backbone-specific because it depends on how often grounding helps that model; under the same question-clustered bootstrap it is 0.698 $[0.670, 0.726]$ on Qwen2.5-7B, 0.738 $[0.699, 0.770]$ on Mistral-7B and 0.701 $[0.664, 0.739]$ on Phi-3.5-mini, so the three intervals overlap heavily and the spread across backbones should not be read as a ranking.

7B, 0.559 on Mistral-7B, 0.505 on Phi-3.5-mini and 0.481 on MuSiQue, every interval containing 0.5. Whatever it knows about utility, it knows through coverage, so the ceiling binds it. The sufficiency signal does not behave this way: its within-coverage AUC is 0.579 $[0.530, 0.629]$, 0.602 $[0.532, 0.663]$ and 0.609 $[0.541, 0.677]$ on the three HotpotQA backbones, and 0.536 $[0.499, 0.576]$ on MuSiQue. The uncovered stratum has too few positives for the same test (as few as 2). This is also why a logistic gate over all three signals reaches 0.704, just above the 0.698 ceiling (Appendix E): one of its inputs is not a coverage estimator.

5 Asking the Model Instead

If the limiting factor is that coverage does not determine utility, the way forward is a signal that is not a function of coverage. Utility depends on whether *this model* needs the passage, so we ask it.

The sufficiency gate shows the passages to the generator and asks whether they contain what is needed to answer, estimating $P(\text{yes})$ by sampling the verdict. It requires no training, no second model, and no retrieval-side statistics.

It is the stronger signal by point estimate on every backbone we tested (Table 2), reaching 0.639–0.724 against 0.602–0.675. Under a question-clustered bootstrap (10,000 replicates, questions as the resampling unit), the sufficiency-minus-reranker difference is +0.104, 95% CI $[+0.021, +0.183]$, p=0.014 on Phi-3.5-mini; +0.049, $[-0.041, +0.137]$, p=0.275 on Mistral-7B; and +0.029, $[-0.030, +0.088]$, p=0.337 on Qwen2.5-7B, where it becomes -0.014 (p=0.674) once the utility label is recomputed from k=5 samples. The 3/3 count is therefore the sign of three point estimates, one of which is distinguishable from zero and one of which reverses under a different labelling.

On a second dataset the order reverses. On the MuSiQue run of Section 4 the sufficiency signal reaches 0.556 $[0.517, 0.599]$ against the reranker's 0.600 $[0.538, 0.661]$, a difference of -0.044 $[-0.118, +0.030]$, p=0.234 — indistinguishable from zero, but negative rather than positive. Counting all four settings, the sufficiency signal is ahead in point estimate in three and behind in one, and separates from zero in one. What the result supports is therefore an existence claim — a signal read off the generator, using no retrieval statistic at all, can be competitive with the standard retrieval-side one — and not a claim that sufficiency is the better gate, nor that it is reliably competitive across datasets.

Its overall AUC does not exceed the coverage ceiling. Because the sufficiency signal carries information within coverage strata (Section 4), the ceiling does not bind it, but in overall AUC it does not beat the ceiling either. On Phi-3.5-mini the gap is +0.005, 95% CI $[-0.060, +0.066]$, p=0.897 (-0.002 under k=5 labels), indistinguishable from zero; on the other two backbones sufficiency sits below the ceiling (-0.059, p=0.010 on Qwen2.5-7B; -0.014, p=0.686 on Mistral-7B). Its independence from coverage is established by the within-stratum test, not by this comparison.

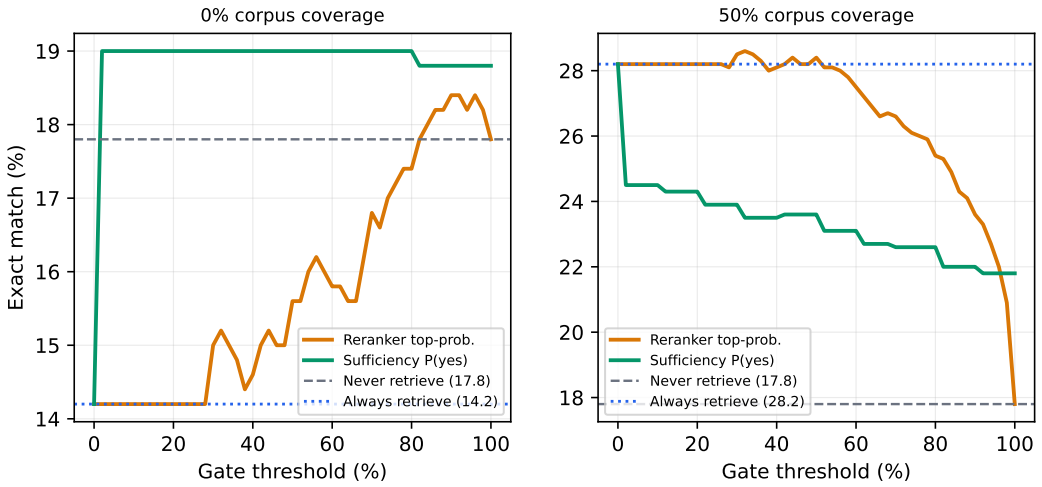


Figure 3: Accuracy against gate threshold at two coverage levels. Left: with no coverage, the sufficiency gate beats both trivial policies across a wide threshold plateau while reranker confidence never meaningfully does. Right: at 50% coverage, always-retrieving is near-optimal and gating only costs.

The sufficiency score is coarse. $P(\text{yes})$ is estimated as the fraction of 10 sampled verdicts at $T{=}1.0$, not read from a logprob, so its resolution is $1/10$ and any true probability below 0.1 collapses to exactly 0. On Qwen2.5-7B greedy, 802 of 1000 instances sit at 0 and the score takes 11 distinct values in total; 51.0% of positive–negative pairs are ties, credited 0.5 under midrank handling. The score is thus close to a binary accept/decline indicator — binarising it moves the AUC to 0.636 from 0.639 — and its AUC should be read as a coarse two-bin statistic rather than a ranking over a continuum. Finer resolution would require more judge samples or a logprob read, neither of which we attempt here.

Gating pays only where coverage is low. Signal quality and threshold placement are different things, so we sweep the threshold rather than report either gate only at its shipped default (Figure 3). The two coverage regimes tell opposite stories, and the contrast is the practical result of this paper.

When the corpus covers nothing, the sufficiency gate is the only policy that helps: it reaches 19.0 EM against 17.8 for NEVER and 14.2 for ALWAYS. Under the same question-clustered bootstrap used throughout (10,000 replicates), that advantage over NEVER is +1.2 EM, 95% CI $[+0.4, +2.2]$, $p{=}0.005$. The benefit appears not to depend on careful tuning: on the observed point-estimate curve every threshold from 2% to 100% beats NEVER, and the gate's shipped threshold already attains the maximum (19.0 EM), so this interval carries no threshold-selection advantage. That span is a description of the point estimates, not a tested range: the bootstrap above establishes only that *some* threshold beats NEVER, which held in 99.7% of the 10,000 replicates, and we compute no interval for the endpoints of the span or for any individual threshold inside it. Reranker confidence, by contrast, barely reaches NEVER at any threshold: its best operating point is 18.4 EM, a difference of +0.6, $[+0.0, +2.4]$, $p{=}0.289$, which is not distinguishable from simply never retrieving.

At 50% coverage the picture inverts. ALWAYS is already close to optimal, and no threshold of either gate beats it by more than noise. This is not a failure of the gates so much as a property of the regime — when retrieval helps more often than it hurts, declining to retrieve can only cost.

The practical reading is therefore narrower than "use a gate": gating is worth its complexity when corpus coverage is genuinely poor, and there the signal to gate on is the model's own assessment, not the retriever's.

Default thresholds still need care. At its shipped threshold the sufficiency gate accepts only 28.4% of GOLD cases, discarding much of the available benefit (Table 1), and the accuracy-optimal threshold differs across backbones. The width of the plateau in Figure 3 suggests this matters less

than it might, though that width is an uncertified feature of the point-estimate curve; an operating point tuned on one model should not be transplanted to another without checking. We report AUC throughout precisely because it separates signal quality from this tuning question.

6 Two Reproducibility Hazards

We record two failures encountered here because both are silent, both affect local-inference evaluation generally, and one of them briefly produced a plausible but entirely spurious result in our own analysis. Appendix D gives both in full.

Backend disagreement. CPU and GPU builds of the same quantised checkpoint disagree. On an identical 100-question set with greedy decoding, CPU scored 73.0% and GPU 68.0%; 95% of predictions matched and all 5 scoring disagreements favoured CPU. A 5.0-point swing attributable to backend alone is comparable to the effects such papers study, yet evaluations rarely state which backend produced their numbers.

A silent all-zero logit buffer. We initially read $P(\text{yes})$ from the inference library's per-token score buffer, which in its default configuration is never written and is returned as allocated — in practice all zeros. The gate therefore computed $\mathrm{softmax}(0, 0) = 0.5$ for every query, accepted unconditionally, and produced a signal scoring just above chance. Nothing raised an error: the result looked like a well-behaved negative finding and we initially read it as one. We now estimate $P(\text{yes})$ by sampling the model's verdict, which uses only the public generation API. Any pipeline reading raw logit buffers should assert that they are non-constant.

7 Discussion

Our results point in a specific direction. Selective retrieval is worth pursuing — the oracle gap in Table 1 shows real headroom — but the quantity to estimate is not retrieval confidence.

Predicting utility means predicting whether *this* model, on *this* question, does better with the passage than without. That depends more on the model's parametric knowledge than on the passage's quality, and the retrieval pipeline has no visibility into it. This explains our main finding — why reranker confidence is a good coverage detector (AUC 0.734) and a poor utility gate (0.602–0.675) — and makes it unsurprising that a signal read off the generator, using no retrieval statistics at all, is competitive with it.

The natural next step is to strengthen the self-assessment signal rather than the retrieval one — for instance by eliciting the model's answer confidence *without* the passage and gating on the difference, which would target utility directly rather than through the proxy of sufficiency.

Limitations. We use two datasets, one retrieval stack, and three backbones of similar scale; the second dataset is run on one backbone only. The within-coverage test that decides which signals the ceiling binds is possible only in the covered stratum, where positives are plentiful. Equation (2) is exact and carries to any balanced coverage design, but its inputs do not: p_1 and p_0 are properties of a dataset–model pair, so the ceiling *values* are estimates for this regime rather than universal constants — as the HotpotQA-to-MuSiQue shift shows directly. What transfers is the decomposition, not the number. HotpotQA's distractors are adversarially selected, making NoGold harder than a typical uncovered query. The utility label compares single-sample outcomes, so it inherits decoding variance; we report a multi-sample robustness check in Appendix E. Finally, we study inference-time gates over a frozen pipeline and say nothing about jointly trained retrieval.

8 Conclusion

Selective retrieval is usually framed as a confidence problem and is better understood as utility prediction. Knowing coverage exactly caps utility prediction at AUC 0.698 on HotpotQA and 0.757 on MuSiQue. The cap binds every signal informative only through coverage, and it decomposes into two rates a practitioner can measure without building a gate. Reranker confidence is such a signal: it detects coverage (AUC 0.734), predicts utility at 0.602–0.675, and carries no detectable

utility information within the covered set. Asking the generator whether the evidence suffices does carry that information on HotpotQA, though its overall advantage is significant on one backbone and absent on MuSiQue; we offer it as an existence proof, not as a better gate. The durable conclusions are the bound and the test of which signals it binds.

REPRODUCIBILITY STATEMENT

Every quantity reported in this paper is generated programmatically from locked JSON result files under `data/results/` and injected as a LaTeX macro; no number is transcribed by hand. Appendix A lists the exact commands that reproduce the result files, the macro generators, and the figures, together with the two shared gate modules and their unit tests. All inferential statements use the question-clustered bootstrap of `scripts/bootstrap_analysis.py` and `scripts/bootstrap_operating_point.py` (10,000 replicates, fixed seed, questions as the resampling unit, so that the paired GOLD/NOGOLD rows of a question are resampled together); both scripts are deterministic given the seed and re-emit `data/results/statistics.json` unchanged. Section 6 documents two backend-level reproducibility hazards we encountered, including the exact CPU/GPU accuracy gap on an identical question set, because both are silent failures that would otherwise be invisible to a replicator. The whole study is cheap to rerun: the 7 runs reported here spent 1.8 GPU-hours inside model calls across 2300 questions on a single consumer GPU (NVIDIA GeForce RTX 4060 Laptop, 8 GB), with no training or fine-tuning. That figure sums the per-call latencies recorded in the result files and counts generation, reranking and judge calls only, so it is a lower bound on total project compute.

ETHICS STATEMENT

This work involves no human subjects, no personal or private data, and no new data collection. Both datasets are public research benchmarks used within their intended scope, and all models are publicly released checkpoints run locally.

The result most relevant to the Code of Ethics is one about research integrity rather than about our own method. Appendix C reports that a corpus assembled from benchmark-adjacent material silently contained documents pairing evaluation questions with their answers, and Section 6 reports two silent failure modes — a backend accuracy gap and an all-zero logit buffer — one of which produced a plausible but spurious result in our own analysis before we caught it. We report all three because each inflates precisely the kind of number this literature publishes, and because we would not have found them by reading our own results. Our primary contribution is a negative one, a ceiling, and we have tried throughout to state where the evidence stops rather than where the narrative would prefer it to continue.

AI USE STATEMENT

Large language models are an object of study in this work: the sufficiency gate of Section 5 elicits a verdict from the generator under evaluation, and every backbone named in Section 3 is run locally as part of the experiments. That use is the subject matter of the paper and is described in full there. This statement concerns our use of generative AI as a research tool.

We used generative AI as a research tool in three ways. First, to aid and polish writing: editing drafts for clarity and concision and suggesting wording. Second, in research ideation and execution: developing and refining research ideas, and carrying out the experiments and analyses, including implementing the analysis code. Third, to draft sections of this paper, which we then revised.

We have reviewed all AI-assisted work. Every number reported in this paper is emitted by the committed scripts from the result files described in the Reproducibility Statement, so each can be regenerated and checked independently of any AI tool. We take full responsibility for the final content of this work, including all text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *PMLR*, pp. 1321–1330, 2017.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *Proceedings of ICML*, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of EACL*, 2021.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 2024.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *International Conference on Learning Representations*, 2025.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020.

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*, 2023.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, 2023.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of NAACL*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, 2018.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *International Conference on Learning Representations*, 2024.

A Reproducibility

Every number is generated programmatically from locked JSON result files; none is transcribed by hand.

```
python scripts/run_coverage_experiment.py --n 500
--n-gpu-layers -1
python scripts/run_fixed_corpus_experiment.py --n 300
--n-gpu-layers -1
python scripts/generate_coverage_macros.py
python scripts/generate_backbone_table.py
python scripts/generate_fixed_corpus_macros.py
python scripts/generate_coverage_figures.py
python scripts/analyze_gate_signals.py
python scripts/bootstrap_analysis.py
python scripts/bootstrap_operating_point.py
python scripts/generate_backend_hazard_macros.py
python scripts/ceiling_closed_form.py
python scripts/run_coverage_experiment.py --dataset musique
--n 500 --n-gpu-layers -1
python scripts/musique_replication.py
python scripts/within_coverage_auc.py
```

The two gates under study live in shared modules, `src/policy_gate.py` and `src/sufficiency_gate.py`, imported by the experiment rather than reimplemented inside it, and covered by unit tests that drive every decision branch with an injected generator.

B The Ceiling for Scores Uninformative Given Coverage

Let $c \in \{0,1\}$ be coverage, $y \in \{0,1\}$ utility, and s any score with $s \perp y \mid c$. Write F_1, F_0 for the distribution of s given $c{=}1$ and $c{=}0$; by conditional independence these do not depend on y. Let $q = P(S_1 > S_0) + \frac{1}{2}P(S_1 = S_0)$ for independent $S_1 \sim F_1$, $S_0 \sim F_0$, and let a and b be as in Section 4. Drawing a positive and a negative independently and conditioning on their coverage bits,

$$\mathrm{AUC}(s) = a(1-b)\,q + (1-a)b\,(1-q) + \tfrac{1}{2}\left[ab + (1-a)(1-b)\right],$$

Policy	Accuracy (%)	Accept rate (%)
Never retrieve	72.0	—
Always retrieve	70.3	100
Reranker-confidence gate	71.3	31.0
Sufficiency gate	72.0	0.3
Oracle utility gate	78.3	—

Table 3: MMLU over a standing index that does not cover it, n=300. Retrieval costs 1.7 points. The sufficiency gate declines almost everything and thereby matches the best trivial policy exactly; the reranker gate accepts 31.0% and pays for it.

since pairs with equal coverage bits are exchangeable and contribute $\frac{1}{2}$. This is linear in $q \in [0, 1]$; when $a \geq b$ it is maximised at $q = 1$, where it equals Equation (1). So no score that is uninformative about utility given coverage can exceed the coverage ceiling (or, when $a < b$, its reflection $1-\text{AUC}$). A score that violates $s \perp y \mid c$ is not covered by this argument, which is why Section 4 tests the condition empirically.

C A Second Regime: A Standing Corpus

This appendix is a supporting result. The paper's claims rest on the controlled coverage experiment; here we check only whether the same policies behave sensibly in a deployment regime where coverage cannot be controlled.

The coverage experiment supplies each question's candidates per query, which makes coverage controllable but keeps the retriever looking in roughly the right place. Deployed systems query a standing index that may simply not contain the answer, and a gate's task there is different: recognise that *nothing* in a large corpus is relevant, rather than choose among ten candidates of which two are gold.

We therefore evaluate the same four policies on 300 MMLU (Hendrycks et al., 2021) questions against this project's fixed 1000-chunk index, built from Wikipedia passages and a general machine-learning guide — a corpus that covers almost none of MMLU's specialised subjects. There is no coverage ground truth here, which is the price of realism, but utility labels need none.

The regime behaves as the coverage–utility framing predicts. Grounding helps on only 6.3% of questions, so retrieval is a net loss: ALWAYS scores 70.3% against NEVER's 72.0%. The reranker gate accepts 31.0% of queries — it cannot tell that the corpus is irrelevant — and lands between the two. The sufficiency gate accepts 0.3% and therefore reproduces NEVER exactly, which is the correct behaviour when retrieval has nothing to offer. An oracle would still reach 78.3%, leaving 6.3 points on the table.

The right behaviour here is not a discriminating signal. It would be easy to read the previous paragraph as showing that the sufficiency signal ranks these queries well. It does not. At predicting utility on this set it reaches AUC 0.498 — chance — and reranker top-probability reaches 0.428, below chance. The sufficiency figure is degenerate rather than merely uninformative: in this regime the gate returns the same probability for 99.7% of these 300 questions, so 0.498 is close to what crediting ties at 0.5 returns for a constant score, and is not a measurement of how the signal orders queries. Neither AUC in this paragraph is bootstrapped, and with utility positive on only 6.3% of 300 questions we do not claim either is distinguishable from chance or from the other. The sufficiency gate matches NEVER because it declines 99.7% of queries and NEVER is the best policy available when grounding helps on 6.3% of questions, not because it identifies *which* queries retrieval would help. The oracle gap of 6.3 points is exactly the part neither signal claims. The correct summary of this regime is therefore about accept rates and their consequences: one gate is calibrated to the corpus's irrelevance and the other is not. Any reading of it as evidence that self-assessment discriminates utility here is unsupported by 0.498.

This is the regime where the two gates differ most starkly in behaviour, and it is also the regime deployed systems actually face.

Answer leakage in the index. Auditing the corpus for this experiment, we found 57 source documents — one per MMLU subject, 285 question–answer pairs in total — generated from MMLU's dev split that embed questions together with their correct answers ("...Options: A) 0 B) 1 ...Correct answer: 1"). Their chunks are topically close to MMLU queries and were retrieved often: excluding them removed 1837 retrieved passages across 300 questions. We filter them, and report the count rather than silently dropping it. We flag this because corpora assembled from benchmark-adjacent material are easy to build by accident and hand the model the answer rather than evidence — an evaluation failure that inflates exactly the retrieval-benefit numbers such papers report.

D The Two Reproducibility Hazards in Full

Backend disagreement. The two runs compared in Section 6 are variant A (no retrieval) of the same controlled ablation, executed on the same 100-question set on the same day, differing only in whether llama.cpp was built with GPU offload. 5 questions were scored correct by the CPU build and not the GPU build, and 0 the other way, so the gap is not sampling noise around a shared decode but a one-directional divergence. Both runs use greedy decoding, so no sampling seed is involved. The comparison is regenerated by `scripts/generate_backend_hazard_macros.py`.

A silent all-zero logit buffer. The defect surfaced only when a longer prompt on a different backbone exceeded the buffer's row count and finally threw. Until then every symptom was consistent with a real but weak signal: the gate accepted unconditionally, its AUC sat just above chance, and accuracy differences against the trivial policies were small but not absurd. We report it because a constant score is indistinguishable from an uninformative one by AUC alone — both sit near 0.5 — and because the failure is silent by construction: no exception, no warning, and a plausible number at the end. The tie fractions quoted in Section 5 are reported partly as a standing check against this class of defect.

E Multi-Sample Robustness of the Utility Label

Under greedy decoding the utility label compares two single samples, so one may reasonably ask whether utility outcomes are decoding accidents — in which case they would be unpredictable by construction and our ceiling would reflect label noise rather than the task.

We re-ran the experiment on 300 questions with $k{=}5$ samples per condition at temperature 0.7, defining utility on *expected* accuracy, with the sufficiency score estimated from sampled verdicts as described in Section 6.

Utility-prediction AUC	Greedy, $n{=}500$	$k{=}5$, $n{=}300$
Sufficiency $P(\text{yes})$	0.639	0.605
Reranker margin	0.594	0.603
Reranker top-probability	0.610	0.619
Supervised (logistic)	0.704	0.680
Perfect coverage detector	0.698	0.697

Table 4: The coverage ceiling is robust to how accuracy is measured. The relative ordering of the two gating signals is not. The supervised gate is a 5-fold cross-validated logistic regression on the reranker top-probability, reranker margin and sufficiency scores; because one input is not a coverage estimator, it is not bound by the ceiling.

The ceiling holds. The coverage ceiling is essentially identical under both labellings (0.698 greedy versus 0.697 multi-sample), as is the reranker signal. Averaging raises the base rate of useful grounding (34.3% of covered questions against 29.4% greedy), which is expected — grounding that helps on average can still fail on a single decode — but it does not make utility more predictable from coverage. The ceiling is a property of the prediction problem.

The sufficiency advantage does not survive on this backbone. Under greedy labels on Qwen2.5-7B the sufficiency signal leads (0.639 versus 0.610); under multi-sample labels the two are effec-

tively tied and the ordering reverses (0.605 versus 0.619). The cross-backbone results of Section 5 use greedy labels throughout, so the 3/3 margin reported there should be read with this in mind: the point estimate is largest on Phi-3.5-mini and Mistral-7B, and on Qwen2.5-7B it is small enough to depend on how utility is defined.

We re-ran the cross-backbone comparison under multi-sample labels (k=5, T=0.7, n=300 questions per backbone; raw runs in `data/results/coverage_backbones_ms/`, analysed by `scripts/bootstrap_analysis.py`). The greedy ordering is unchanged on the other two backbones: sufficiency minus reranker is +0.051 on Mistral-7B and +0.101 on Phi-3.5-mini, against +0.049 and +0.104 under greedy labels. The reversal is therefore specific to Qwen2.5-7B rather than to multi-sample labelling. Under a question-clustered bootstrap, however, the sufficiency advantage is distinguishable from zero on Phi-3.5-mini alone (greedy p=0.014; Qwen2.5-7B p=0.337, Mistral-7B p=0.275), so the 3/3 count reported above is the sign of three point estimates and not three significant differences. The practical claim of Appendix C — that the sufficiency gate correctly declines when a standing corpus is irrelevant while reranker confidence does not — is a difference in behaviour rather than in AUC, and is unaffected.