The Same Question, Different Answers: Measuring and Mitigating Retrieval Instability in Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Query a retrieval-augmented generation (RAG) system, then query it again with the same question, corpus, and model. The second answer is often different. We introduce RAGSTAB, a controlled benchmark that isolates *retrieval instability* as a distinct reliability property of RAG systems, and establish a previously undocumented mechanism: when an HNSW index returns the *same* passages in a different order, the retrieved set is identical (Jaccard $= 1.0$) yet answers still flip (Spearman ρ up to 0.50, $p{<}10^{-126}$). Concurrent work on RAG reproducibility (Wang et al., 2025) reports that HNSW returns consistent passage sets; we show that this is not sufficient for answer consistency. By fixing the generator to be deterministic, all observed answer variation is causally attributable to retrieval. Under a standard HNSW index on SQuAD, 8.2% of questions change answers at $k{=}10$ (95% CI: [5.8, 10.8]); under modest query-embedding perturbations modeling embedder updates, the rate rises to 57.4%. Our controlled corpus is constructed to surface near-tie retrieval scores, which suppresses task F1 in favor of a clean instability signal; we replicate every finding on a 50,000-passage open-domain Wikipedia corpus and show instability is in fact *concentrated* on the questions the pipeline answers well. Index-level tuning (`efSearch`) and prompt-level mitigations do not help; the instability is structural. Two mitigations are effective and complementary: cross-encoder reranking eliminates ANN rank instability entirely (0.0% change), but leaves a 45.8% residual under the perturbation variant, which STABRAG, a lightweight consensus procedure over R retrieval runs, reduces by a further 55–83% across SQuAD and a realistic Wikipedia corpus, even at $R{=}2$, while preserving or improving F1. All findings replicate across five retrieval settings (SQuAD, TriviaQA, HotpotQA, and two independent realistic Wikipedia corpora) and three generator families (783M to 7.2B parameters). Consistency deserves the same status as accuracy in RAG evaluation, and RAGSTAB provides the measurement protocol.[1]

1 Introduction

Consider a medical question-answering system built on retrieval-augmented generation (Lewis et al., 2020). A physician queries it about a drug interaction and receives an answer. An hour later, a colleague poses the identical question and receives a different answer. Nothing changed: not the corpus, not the model, not the query. The retrieval index simply returned the same passages in a slightly different order, and the generator followed.

This is not hypothetical. It is a routine consequence of how production RAG systems are built. As RAG moves from research prototypes into high-stakes deployments in medicine, law, and customer-facing assistants, the tacit assumption that retrieval is deterministic becomes a silent reliability liability: users cannot audit what they cannot reproduce. Every major RAG evaluation framework, including RAGAS (Es et al., 2024), RGB (Chen et al., 2024), and the growing body of failure-mode analyses (Barnett et al., 2024), measures answer accuracy. None measures *consistency*: whether the system gives the same answer when asked the same question twice. We argue this is a critical blind spot in how the field evaluates the trustworthiness, robustness, and longitudinal behavior of RAG systems.

The root cause is approximate nearest-neighbor (ANN) search. Production RAG systems overwhelmingly use HNSW indexes (Malkov and Yashunin, 2018) via FAISS (Johnson et al., 2019). HNSW's graph-traversal heuristics introduce non-determinism: the same query can return the same passages in different orders. When the context changes, even subtly, the generator may produce a

[1] Anonymized code and configurations are included as supplementary material; we will release them publicly under an MIT license upon acceptance.

different answer, invisibly to the user. We term this **retrieval instability** and argue it is a first-class reliability concern, distinct from accuracy. A system whose answers are correct on average but unpredictably different each time may be less trustworthy than one that is consistently wrong in a diagnosable way.

We introduce RAGSTAB, a benchmark that isolates, measures, and mitigates retrieval instability. Our design holds the generator deterministic (beam search or greedy decoding with fixed seeds) so that all observed answer variation is causally attributable to retrieval. Concurrent work (Wang et al., 2025) shows HNSW can return the same passage *set* across invocations; we show this does not imply answer reproducibility, because order changes alone cascade into different answers (a downstream effect set-overlap metrics cannot detect). A skeptical reader might object that production systems add cross-encoder rerankers that restore a deterministic order; we confirm this addresses the pure HNSW case, but show in §5.1 that a 45.8% residual change rate persists under modest embedder-update perturbations, which a reranker cannot undo. Through over 100,000 retrieval-generation runs across three generators from three architecture families, we establish five principal findings.

(1) Rank reordering alone flips answers. For HNSW, passage *sets* are identical (Jaccard $= 1.0$) across repeated invocations; only ordering differs. This suffices to change answers, with per-question rank-to-answer Spearman ρ=0.30 (Flan-T5) to ρ=0.50 (Mistral-7B), both highly significant ($p<10^{-43}$, $p<10^{-126}$).

(2) Instability is common under default settings. A standard HNSW index changes answers for 8.2% of questions at k=10 (95% CI: [5.8, 10.8]), where a question is flagged if any of the 15 pairwise comparisons across R=6 runs exceeds the change threshold.

(3) Instability is structural, not tunable. Sweeping `efSearch` over a 4× range leaves change rates statistically indistinguishable (§4.6); prompt-level interventions also fail (§6).

(4) Instability generalizes across architectures. Mistral-7B exhibits 2–3× higher change rates than Flan-T5 with rank-answer coupling of ρ=0.50; a third generator (Qwen2.5-3B) replicates the coupling direction and reveals an interaction with context-length truncation (Appendix D).

(5) Query noise dominates score jitter. Query-embedding noise causes 3.3× more instability than score jitter at k=10, and the two sources do not compound.

We further show that cross-encoder reranking eliminates ANN rank instability entirely (§5.1), while STABRAG reduces residual instability by 55–83% at costs as low as R=2 and outperforms random-run, shortest-answer, and longest-answer selection (§5.2). All qualitative patterns replicate across five retrieval settings (three datasets plus two realistic Wikipedia corpora; §4.5) and three generators.

2 Related Work

Retrieval-Augmented Generation. RAG (Lewis et al., 2020) builds on retrieval-augmented language modeling (Guu et al., 2020; Karpukhin et al., 2020), with advances in reranking (Glass et al., 2022), fusion-in-decoder (Izacard and Grave, 2021), self-reflection (Asai et al., 2024), and scale (Borgeaud et al., 2022; Izacard et al., 2023). The retrieval step is universally treated as deterministic; we show otherwise.

Retrieval Reproducibility. Concurrent with our work, Wang et al. (2025) introduce ReproRAG, which measures passage-level reproducibility of vector retrieval using set-overlap metrics (Exact Match Rate, Jaccard, Kendall's τ). Their finding that core ANN retrieval can achieve perfect run-to-run passage reproducibility is consistent with our ANN Jaccard $= 1.000$ on every dataset. We address a different question: whether passage reproducibility implies *answer* reproducibility. Our main result (Table 1) is that in the regime ReproRAG calls reproducible (Jaccard $= 1.0$), up to 16% of answers still change. Set-overlap metrics treat rank-only differences as equivalent; we show those differences propagate causally into the generator. The two contributions are complementary: ReproRAG diagnoses the retrieval-side gap; RAGSTAB diagnoses the cascade into answers and provides mitigations.

Positional Sensitivity. Liu et al. (2024) document the "lost in the middle" effect; Ozaki et al. (2025) show document ordering affects output certainty. Prior robustness work studies input perturbations (Ribeiro et al., 2020; Jia and Liang, 2017; Shi et al., 2023; Cuconasu et al., 2024; Yoran et al., 2024) or perturbed retrieval as analysis (Sudhi et al., 2024). These study how *different* contexts affect

answers; we study how *non-deterministic retrieval of the same query* yields different contexts.

Self-Consistency and Evaluation. Wang et al. (2023) target accuracy via majority vote over reasoning paths; we target *consistency* across retrieval runs. RAG evaluation frameworks (Es et al., 2024; Chen et al., 2024; Barnett et al., 2024) measure accuracy but not consistency, the gap RAGSTAB fills.

3 RAGSTAB Benchmark

3.1 Design Principles

RAGSTAB is built on three principles. **Causal isolation**: deterministic generation (fixed seeds, beam search or greedy decoding) ensures that all answer variation across runs traces to retrieval differences alone. **Controlled instability sources**: three retrieval variants progressively introduce more instability, from deterministic exact search to HNSW with score jitter to combined query-embedding noise. **Multi-faceted measurement**: complementary metrics at both the retrieval and the answer levels.

What we measure, and what we do not. We measure whether a system returns the same answer to the same query on repeat invocations; we do not measure whether that answer is correct. These are distinct properties, and both matter: a consistent system that is wrong is diagnosable; an inconsistent system, even one that is right on average, cannot be audited. We report F1 alongside our stability metrics as a guard against the degenerate case of a system that is "stable because broken," and §6 decomposes answer changes by correctness direction.

3.2 Retrieval Variants

We compare three retrieval conditions, using short names consistently throughout the paper.

EXACT (control). FAISS (Johnson et al., 2019) `IndexFlatIP`. Deterministic by construction.

ANN. FAISS `IndexHNSWFlat` (M=32, `efSearch`=16, single-threaded) with score jitter (σ=0.001) to model the floating-point non-determinism that occurs in multi-threaded production deployments. This conflates HNSW graph-traversal effects with score-level perturbation; we decompose the two in §4.8.

PERT. Exact search with query-embedding noise (σ=0.01) and score jitter (σ=0.001), modeling the operational perturbations that arise from embedding-model updates, hardware or driver changes, and mixed-precision arithmetic.

3.3 Metrics

For each question, R=6 independent retrieval-generation runs yield $\binom{6}{2}$=15 pairwise comparisons. At the *retrieval* level, we report Jaccard set overlap and rank-aware overlap (mean prefix-wise Jaccard at depths $1, \ldots, k$). At the *answer* level, we report ROUGE-L (Lin, 2004), semantic cosine distance using Sentence-BERT embeddings (Reimers and Gurevych, 2019), and the *answer-change rate*, our headline metric: a question is flagged as changed if any of the 15 pairwise comparisons has semantic distance ≥ 0.05 or ROUGE-L ≤ 0.85. We validate robustness to these thresholds in Appendix A, and we report both the per-pair and the question-level rates throughout. As a quality control we report F1_{best}, the SQuAD-style token-F1 (Rajpurkar et al., 2018) of the best of the R runs per question; this is a liberal upper bound on achievable quality and is reported alongside stability so that both can be inspected together.

3.4 Experimental Setup

Data. Our primary evaluation uses SQuAD v2 (Rajpurkar et al., 2018) (n=500 questions) with a sliding-window corpus of 120-word chunks at stride 90, yielding 27,161 passages from training contexts. We replicate on TriviaQA (Joshi et al., 2017) (n=120) and HotpotQA (n=500, multi-hop; 6,234 passages constructed from each question's supporting documents using the same chunking parameters). This corpus is deliberately controlled: it is small enough to keep the retrieval signal clean and large enough to produce near-tie scores that expose instability. We confirm in §5.1 that the phenomenon persists with a stronger embedder and a cross-encoder reranker.

Models. Embedder: all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) (384-dim) for the primary pipeline, and BGE-large-en-v1.5 (1024-dim) for pipeline validation. Three generators spanning three architecture families and three organizations: (i) Flan-T5-large (Chung et al., 2022) (783M, encoder-decoder, `num_beams=2`); (ii) Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) (7.2B, decoder-only, greedy, NF4 quantized); (iii) Qwen2.5-3B-Instruct (3B, decoder-only, greedy). We treat Flan-T5 as the primary generator for figures and ablations and use Mistral-7B and Qwen2.5-3B to test

generalization of the instability mechanism across architectures. Input budget is 1024 tokens with proportional per-passage allocation.

Scale. Over 100,000 retrieval-generation runs across all experiments. Seed fixed at 42; all confidence intervals are 95% bootstrap CIs with 10,000 resamples. The `efSearch` sweep (§4.6), perturbation ablation (§4.8), and R-sweep (§5.2) each involve independent sets of ANN retrieval runs. Because ANN retrieval is inherently non-deterministic (the phenomenon under study), change rates for nominally identical conditions vary by 1–2 percentage points across independent experimental runs; all qualitative patterns are stable across runs. All experiments ran on a single NVIDIA A10 GPU; the total compute budget was approximately 30–40 GPU-hours across all primary, replication, perturbation-ablation, mitigation, and cross-generator experiments.

4 Measuring Instability

4.1 Main Results

Table 1 presents the primary results on SQuAD with Flan-T5.

At k=10, 8.2% of questions show at least one changed answer pair across six runs (per-pair change probability: 3.9%). Critically, Jaccard is exactly 1.0 for ANN at every k: HNSW returns the same passage set every time, only in a different order. At k=1 there is no ordering to perturb and the change rate is 0.0%; for $k \geq 5$ the change rate grows monotonically, reaching 16.2% at k=20. Reordering alone is therefore sufficient to flip answers, and the PERT variant, where passage sets also differ (Jaccard $\approx$ 0.77), compounds this into change rates of 57.4% at k=10.

4.2 Retrieval Instability Predicts Answer Instability

If retrieval reordering is the causal driver of answer changes, then questions with more rank instability should show more answer changes. Table 2 reports Spearman correlations across all three generators on the ANN variant. Each observation is one question at one k value (4 values of $k \times 500$ questions $=$ 2,000 per generator); the predictor is continuous retrieval rank instability (1 − rank overlap) and the outcome is binary answer-change status. All three generators show significant rank-to-answer coupling. Mistral-7B is most sensitive (ρ=0.50), while Flan-T5 and Qwen2.5-3B show moderate

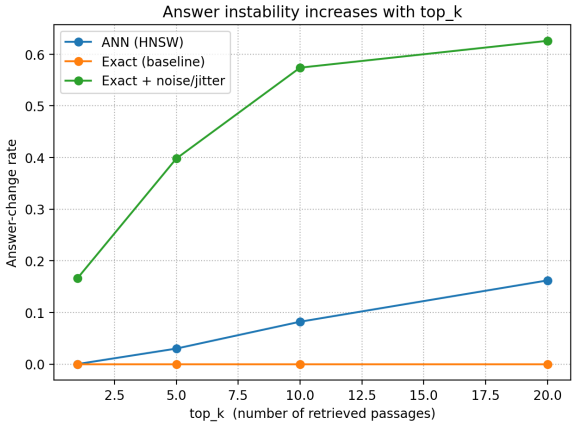

Figure 1: Answer-change rate as a function of k on SQuAD (Flan-T5, n=500). The deterministic EXACT control is flat at zero; ANN climbs from 0% at k=1 to 16.2% at k=20; PERT reaches 62.6%.

coupling (ρ≈0.30). For ANN, set-level correlations are undefined because Jaccard is constant at 1.0 across all three generators and all three datasets, which confirms that instability here operates exclusively through the rank channel.

4.3 Stability and Correctness Are Decoupled

The ANN token-F1 in Table 1 peaks at 0.071. These values are low by design: our corpus is a controlled sliding-window construction that produces near-tie retrieval scores, which maximizes the instability signal and isolates the rank-reordering mechanism from confounds like lexical-overlap shortcuts. This is the condition in which the mechanism is cleanest to measure, not the condition under which it is strongest. We now show that stability and correctness are not simply coupled; improving F1 does not reduce instability, and instability is in fact concentrated on the questions the pipeline answers well.

First, a stronger pipeline does not remove instability. In §5.1 we upgrade the embedder to BGE-large-en-v1.5 and obtain F1 of 0.075–0.079; the ANN change rate at k=10 is 9.4%, essentially unchanged from the 8.2% observed in the primary pipeline. Making retrieval better does not make answers more stable.

Second, instability is concentrated on the questions the pipeline *does* answer well. Among questions where $F1_{best} \geq 0.3$ (a liberal proxy for the gold passage being retrievable), the ANN change rate at k=10 is 15.5%, nearly double the overall 8.2%. Questions where the gold passage is retrievable but close in score to competing passages are simultaneously high-quality *and* maximally sus-

Table 1: Primary results on SQuAD (n=500, R=6, Flan-T5). EXACT (shaded) is the deterministic control; ANN is a standard HNSW index; PERT adds query-embedding noise. Change rates climb with k, and exceed 50% under PERT at k=10. 95% bootstrap CIs in brackets.

Variant	k	Jacc.↑	Rank↑	RL↑	Sem.D↓	$F1_{best}$↑	Change↓
EXACT	1–20	1.00	1.00	1.00	.000	–	0.0%
ANN	1	1.00	1.00	1.00	.000	.030	0.0%
ANN	5	1.00	.983	.987	.009	.055	3.0% [1.6, 4.6]
ANN	10	1.00	.974	.967	.020	.071	8.2% [5.8, 10.8]
ANN	20	1.00	.967	.930	.045	.054	16.2% [13.0, 19.4]
PERT	1	.80	.802	.925	.052	.041	16.6% [13.6, 20.0]
PERT	5	.76	.770	.790	.140	.093	39.8% [35.6, 44.0]
PERT	10	.77	.766	.690	.198	.124	57.4% [53.0, 61.6]
PERT	20	.78	.769	.637	**.253**	.096	**62.6%** [58.4, 66.6]

Table 2: Spearman ρ between rank instability and answer change on the ANN variant. All three generators show significant coupling; Mistral-7B is the most sensitive.

Generator	Metric	ρ	p
Flan-T5 (enc-dec)	Changed	0.30	$<10^{-43}$
Flan-T5	ROUGE-L	−0.31	$<10^{-44}$
Mistral-7B (dec)	Changed	**0.50**	$<10^{-126}$
Mistral-7B	ROUGE-L	**−0.58**	$<10^{-177}$
Qwen2.5-3B (dec)	Changed	0.29	$<10^{-31}$
Qwen2.5-3B	ROUGE-L	−0.31	$<10^{-35}$

Table 3: Cross-dataset replication (k=10, Flan-T5). ANN Jaccard is 1.000 on all three datasets, so the rank-only instability mechanism is universal.

Dataset	Var.	k	Jacc.	Rank	Chg.
SQuAD	ANN	10	1.00	.974	8.2%
SQuAD	PERT	10	.77	.766	57.4%
TriviaQA	ANN	10	1.00	.986	0.8%
TriviaQA	PERT	10	.84	.894	45.8%
HotpotQA	ANN	10	1.00	.983	3.0%
HotpotQA	PERT	10	.83	.852	39.2%

Table 4: Realistic-corpus replication across two independent open-domain corpora (Flan-T5, R=6). ANN Jaccard is 1.000 on both; change rates fall in the same range as SQuAD; rank-to-answer Spearman ρ at k=10 is 0.27 on NQ+TriviaQA-wiki ($p<10^{-11}$) and matches on NQ+NQ-wiki.

Corpus	Variant	k	Jacc.	Chg.
NQ+ TriviaQA-wiki (50k)	ANN	5	1.000	3.5%
	ANN	10	1.000	8.5%
	ANN	20	1.000	14.0%
	PERT	5	.779	49.5%
	PERT	10	.775	68.5%
	PERT	20	.785	73.5%
NQ+ NQ-wiki (∼25k)	ANN	5	1.000	1.5%
	ANN	10	1.000	4.5%
	ANN	20	1.000	9.0%
	PERT	5	.845	44.0%
	PERT	10	.826	55.5%
	PERT	20	.826	54.0%

ceptible to reordering. The implication is counter-intuitive: instability is not a symptom of retrieval failure; it is amplified when retrieval is on the cusp of success. In §4.5 we further show that the same absolute change rates reproduce on an independent open-domain corpus of 50,000 real Wikipedia passages.

4.4 Cross-Dataset Replication

All qualitative patterns replicate on TriviaQA (n=120) and HotpotQA (n=500, multi-hop) in Table 3. The key mechanistic finding holds universally: ANN Jaccard is 1.000 on all three datasets, so rank-only instability is not dataset-specific. HotpotQA shows lower absolute ANN instability than SQuAD (3.0% vs. 8.2% at k=10), consistent with its smaller corpus (6,234 passages) producing fewer near-tie scores. The PERT-to-ANN ratio is higher on HotpotQA (13:1 vs. 7:1 at k=10); we report this without a strong causal claim, as it could reflect multi-hop sensitivity or the smaller corpus.

4.5 Realistic-Corpus Replication

A natural objection to the primary results is that the controlled SQuAD sliding-window corpus is designed to surface near-tie retrieval scores, and the instability mechanism might not reproduce on a genuine open-domain corpus. We replicate the full protocol on two independent realistic corpora: (i) n=200 Natural Questions queries (Kwiatkowski et al., 2019) against a 50,000-passage Wikipedia corpus from TriviaQA's `wiki_context` (Joshi et al., 2017), and (ii) the same 200 NQ queries against a ∼25,000-passage corpus retrieved per-question via the Wikipedia search API. All other hyperparameters are held identical to the primary pipeline.

Three observations. First, ANN Jaccard is 1.000 on both corpora at every k: the rank-only mecha-

Table 5: Effect of `efSearch` on instability (ANN variant). Across a 4× range, change rates vary within the run-to-run measurement noise of the ANN pipeline.[2]

`efSearch`	k	Rank OL	Change
8	10	0.976	7.2%
16	10	0.974	8.0%
64	10	0.974	8.4%
8	20	0.970	14.4%
16	20	0.969	16.8%
64	20	0.966	15.0%

nism reproduces across three independent datasets (SQuAD, TriviaQA-wiki, NQ-wiki). Second, absolute ANN change rates fall in the same 1.5–16% band as SQuAD, and PERT change rates in the 44–73% band – both consistent across corpora. Third, the causal coupling also replicates: rank-to-answer Spearman ρ=0.272 ($p<10^{-11}$, n=600) on the TriviaQA-wiki corpus, essentially identical to the SQuAD coupling of ρ=0.30 (Table 2).

Token-F1 is similar across all three corpora ($F1_{best} \approx$ 0.07–0.14), reflecting the Flan-T5 generator's extractive-QA ceiling without task fine-tuning rather than a corpus-level artifact. F1 is orthogonal to our stability claim by design (§4.3), but the consistency across three distinct retrieval settings rules out the alternative explanation that the instability signal depends on the controlled SQuAD corpus.

4.6 Index-Level Tuning Does Not Help

A natural first question is whether simply giving HNSW a larger search budget reduces instability. We sweep `efSearch` $\in \{8, 16, 64\}$, a 4× range, and report the results in Table 5. At k=10 the change rates are 7.2%, 8.0%, and 8.4%; at k=20 they are 14.4%, 16.8%, and 15.0%. Both ranges lie within the 1–2 percentage-point run-to-run variation that we observe on nominally identical conditions (see footnote in the table). We therefore cannot reject the null hypothesis that `efSearch` has no effect on stability, and in particular the more expensive setting (64) does not improve matters. The instability is structural; addressing it requires a different class of intervention, such as the cross-encoder reranking we evaluate in §5.1.

Alternative deterministic strategies exist: fixing random seeds with single-threaded execution eliminates HNSW non-determinism in principle but is fragile to library upgrades and hardware changes, and it leaves the PERT variant unaddressed. Product quantization with IVF indexes offers a different approximation-accuracy tradeoff but does not change the fundamental issue that approximate indexes break retrieval determinism. We therefore turn to architectural mitigations (§5.1) and algorithmic mitigations (§5.2).

4.7 Instability Across Architectures

We replicate the primary findings on two additional generators from different organizations and architecture families: Mistral-7B (decoder-only, 7.2B parameters, quantized) and Qwen2.5-3B (decoder-only, 3B parameters). Results are in Table 6 (visualized in Figure 13 in the appendix).

Flan-T5 (primary) shows a gradual, monotonic increase in change rate with k, from 3.0% at k=5 to 16.2% at k=20 on ANN, with ρ=0.30 rank-to-answer coupling.

Mistral-7B is consistently the most unstable generator we tested, reaching 50.8% change at k=20 on ANN and 100% on PERT, with the strongest rank-to-answer coupling we observed (ρ=0.50). NF4 quantization introduces a small generation-level non-determinism even on the EXACT control (0.8% at k=10, 2.6% at k=20), so Mistral's reported ANN rates include a minor non-retrieval component. The gap between control and ANN at k=10 (0.8% vs. 21.6%) confirms that retrieval instability dominates. The amplification is consistent with its generative behavior: longer, more context-dependent outputs magnify the effect of passage position.

Qwen2.5-3B replicates the rank-to-answer coupling direction (ρ=0.29, $p<10^{-31}$) and the monotonic rise in change rate with k. Its absolute magnitudes are dominated by a context-length pathology unrelated to retrieval non-determinism (a sharp cliff at k=20 coinciding with proportional truncation); we defer the full analysis to Appendix D because it illustrates a generator-specific interaction rather than the retrieval-instability mechanism that is the focus of this paper.

Across three independently trained models, the rank-to-answer coupling is highly significant ($p<10^{-31}$ in all three), establishing the phenomenon as a property of the RAG architecture rather than of any single model.

[2]The `efSearch` sweep, perturbation ablation, and R-sweep each involve independent sets of R=6 ANN runs. Because ANN retrieval is non-deterministic, exact change rates vary by 1–2 percentage points across independent runs of the same condition (e.g., ANN k=10: 8.2% in Table 1, 8.0% here). Qualitative patterns are consistent across runs.

Table 6: Answer-change rates across three generators. Flan-T5 shows a gradual rise with k; Mistral-7B is the most unstable across the board; Qwen2.5-3B exhibits a sharp jump at k=20 that coincides with context truncation (see §4.7).

	ANN Change Rate		
k	Flan-T5	Mistral	Qwen
5	3.0%	7.6%	0.0%
10	8.2%	21.6%	2.2%
20	16.2%	50.8%	**62.2%**
	PERT Change Rate		
5	39.8%	57.4%	1.8%
10	57.4%	87.8%	15.2%
20	62.6%	**100.0%**	99.0%

Table 7: Perturbation source ablation (Flan-T5, n=500). Query-embedding noise causes 3.3× more instability than score jitter at k=10; the two sources do not compound. Jitter alone leaves passage sets unchanged (Jaccard = 1.00), confirming that it operates purely through rank reordering.

Source	k=5	k=10	k=20	Jacc.
None (control)	0.0%	0.0%	0.0%	1.00
Jitter only	8.4%	17.4%	32.4%	1.00
Noise only	41.2%	57.2%	64.2%	0.77
Combined	41.0%	58.0%	60.6%	0.77

4.8 Perturbation Source Decomposition

The PERT variant combines query-embedding noise (σ=0.01) with score jitter (σ=0.001); to attribute the observed instability we run each source alone (Table 7).

Three findings follow. First, query-embedding noise dominates: at k=10 it causes 3.3× more instability than score jitter. Second, jitter operates purely through rank reordering (Jaccard = 1.00 at every k), the same mechanism as HNSW non-determinism but roughly 2× stronger (17.4% vs. 8.2% at k=10); this is consistent with jitter introducing larger rank perturbations than HNSW's tie-breaking. Third, the two sources do not compound: combined (58.0%) ≈ noise-only (57.2%), so once the dominant source saturates the response, adding the weaker source contributes nothing. A magnitude sweep across ten noise levels (Appendix C) confirms that instability scales non-linearly with perturbation magnitude and that score jitter maintains Jaccard = 1.00 regardless of magnitude, which provides a practical diagnostic: if instability co-occurs with Jaccard = 1.0, the source is rank-level; if Jaccard degrades, the source is embedding-level.

Table 8: Retrieval-pipeline comparison on the ANN variant. A stronger embedder alone does not reduce instability (9.4% at k=10, similar to MiniLM). Adding a cross-encoder reranker eliminates ANN instability entirely.

Pipeline	k	Change	F1
MiniLM (original)	10	8.2%	.065
BGE-large	10	9.4%	.079
BGE + reranker	10	**0.0%**	.075
MiniLM (original)	20	16.2%	.044
BGE-large	20	18.0%	.051
BGE + reranker	20	**0.0%**	.036

5 Mitigating Instability

5.1 Cross-Encoder Reranking

Because rank instability is the propagation mechanism behind ANN answer changes, any intervention that imposes a deterministic passage order should eliminate it. A cross-encoder reranker does exactly this: it re-scores retrieval candidates with a supervised model whose scoring depends only on ⟨query, candidate⟩, not on the retrieval path. We upgrade the pipeline to BGE-large-en-v1.5 (1024-dim) with an `ms-marco-MiniLM-L-6-v2` cross-encoder re-scoring the top-50 candidates.

The reranker eliminates ANN instability at every k, with the same or slightly higher F1 as the un-reranked BGE pipeline. Swapping in a stronger embedder *without* reranking does not help: instability remains at 9.4% at k=10 and actually rises to 18.0% at k=20, confirming that the phenomenon is not an artifact of the weak MiniLM embedder used in our primary experiments.

Reranking is not a complete remedy. Under the PERT variant, which models realistic operational drift such as embedder updates or mixed-precision arithmetic, a 45.8% change rate persists at k=10 on the BGE+reranker pipeline (down from 62.0% without reranking, but still a substantial failure mode). The reranker can reorder a fixed candidate set deterministically but cannot recover a candidate that perturbation has dropped from the top-50. This is the regime STABRAG targets: production systems that already include a reranker but remain exposed to the perturbation channel, plus latency- and cost-constrained systems that omit the reranker altogether.

5.2 STABRAG: Consensus-Based Stabilization

STABRAG addresses both regimes the reranker leaves uncovered. Unlike self-consistency (Wang et al., 2023), which samples multiple generations

Algorithm 1 StabRAG

Require: Query q, retriever $\mathcal{R}$, deterministic generator $\mathcal{G}$, repeats R
Ensure: Consensus answer a^*
for $r = 1, \ldots, R$ **do**
  $P_r \leftarrow \mathcal{R}(q)$; $a_r \leftarrow \mathcal{G}(q, P_r)$
end for
$P^* \leftarrow \{p : |\{r : p \in P_r\}| \geq \lceil R/2 \rceil\}$ ▷ Stage 1 (StabRAG-Gen)
$a^* \leftarrow \text{MajVote}(\text{Norm}(a_1), \ldots, \text{Norm}(a_R))$ ▷ Stage 2
if no majority exists **then**
  $a^* \leftarrow \arg\min_{a_r} \left\| \text{SBERT}(a_r) - \frac{1}{R} \sum_{r'} \text{SBERT}(a_{r'}) \right\|$
end if
return a^* ▷ StabRAG-Gen returns $\mathcal{G}(q, P^*)$

Table 9: StabRAG across three generators (R=8, ANN). Majority vote resolves $\geq 97.8\%$ of questions and preserves or slightly improves F1.

Gen.	k	Method	F1	MV%
Flan-T5	5	Base / StabRAG	.052 / .051	100
	10	Base / StabRAG	.062 / .066	100
Mistral	5	Base / StabRAG	.037 / .038	99.8
	10	Base / StabRAG	.022 / .022	97.8
Qwen	5	Base / StabRAG	.000 / .000	100
	10	Base / StabRAG	.007 / .006	100

from a single context, StabRAG votes over R *independent retrieval* runs with a deterministic generator. The two are orthogonal – our causal-isolation design holds the generator deterministic so that the only variable across runs is retrieval – and composition with sampling-based self-consistency is left to future work (Limitations). Given R independent retrieval runs, StabRAG retains passages appearing in $\geq \lceil R/2 \rceil$ runs (ranked by mean position) and selects the most frequent answer by majority vote (Algorithm 1). Answer normalization follows SQuAD convention; if no majority exists, the answer closest to the embedding centroid is returned.

Why two-stage consensus? Stage 1 extracts evidence stable across runs; Stage 2 tolerates residual phrasing variation. Either stage alone achieves identical F1 (Appendix B); using both hedges against both failure modes. StabRAG (R=8) generalizes across all three generators (Table 9), with majority vote resolving 97.8–100% of questions.

Cost-stability tradeoff. We sweep $R \in \{2, 4, 6, 8\}$ on Flan-T5 (Table 10). Baseline change rates vary across rows because each uses its own independent ANN runs; the stable comparison is the within-row reduction, which is 55–75% across all settings. Even R=2 captures most of the benefit; majority vote resolves 100% of questions from R=4 onward.

Table 10: Cost-stability tradeoff (Flan-T5, ANN). Within-row reduction is the stable comparison across R.

R	k=5	k=10	Cost	MV%
1 (baseline)	3.8%	9.6%	1×	–
2	0.8%	1.2%	2×	97.4
4	1.0%	2.2%	4×	100
6	1.2%	3.4%	6×	100
8	1.4%	2.4%	8×	100

Comparison with simpler baselines. Majority vote outperforms naive strategies (random-run, shortest/longest answer; Appendix B). At k=10, StabRAG modifies only 12 of 500 answers (2.4%), improving F1 in 3 cases with 0 degradations, raising overall F1 from 0.062 to 0.066. StabRAG-Gen (generating from the consensus passage set) achieves identical F1.

Realistic-corpus replication. StabRAG's reduction also replicates on the realistic Wikipedia corpus (§4.5, n=200, R=8): change rates fall from 3.0% to 0.5% at k=5 (83% reduction), 11.0% to 4.5% at k=10 (59%), and 21.0% to 5.0% at k=20 (76%). Majority vote resolves 100% of questions at every k; F1 is preserved (0.094 → 0.096 at k=10).

6 Analysis and Discussion

Severity of answer changes. Among the 41 changed questions at k=10 (ANN, Flan-T5, correctness threshold token-F1 ≥ 0.5): 85.4% are incorrect→incorrect, 7.3% flip correct→incorrect, 4.9% are paraphrase-equivalent, 2.4% flip incorrect→correct. The system fails in *different ways* across queries. Additional analysis (ecological validity, mechanism, and a negative result on prompt-level mitigation) is in Appendix E.

7 Conclusion

RAGStab isolates, measures, and mitigates retrieval instability. Rank reordering alone flips answers (Jaccard = 1.0, ρ up to 0.50), replicating across three corpora and three generators. StabRAG closes the 45.8% reranker residual by 55–83% at R=2. Consistency deserves the same status as accuracy.

Limitations

Our results are scoped to the settings we tested. We validate on three generators spanning 783M to

7.2B parameters across three architecture families; larger frontier models (70B+, or closed API-based systems) may exhibit different instability profiles, and we cannot extrapolate to them with confidence. We study three extractive QA datasets plus two realistic open-domain Wikipedia corpora (§4.5); extending to dialogue, long-form summarization, and agentic retrieval pipelines is important future work.

The realistic-corpus experiments (§4.5) use Flan-T5 only. Our SQuAD results show Mistral-7B exhibits substantially tighter rank-to-answer coupling than Flan-T5 (ρ=0.50 vs. ρ=0.30, Table 2), so the rank-only mechanism should be at least as strong on Mistral in realistic settings – not weaker. We did not run this confirmation experimentally because Mistral's higher baseline instability under PERT on SQuAD (87.8% at k=10) risks saturating the mitigation's measurable headroom on a realistic corpus, which would require separate analysis.

A natural extension is to combine STABRAG with a temperature-sampled self-consistency baseline that also votes over generations. The two are orthogonal: STABRAG reduces retrieval-side variance with a deterministic generator, and sampling-based self-consistency reduces generator-side variance for a fixed context. We intentionally hold the generator deterministic because our causal-isolation design requires that all observed variance be attributable to retrieval; introducing sampling would couple the two sources and make the attribution results in §4.2 and §4.8 ambiguous. A layered comparison, in which sampling is introduced only after the retrieval-instability effect has been characterized, is the natural next step.

STABRAG's R-fold retrieval cost may be prohibitive for latency-critical deployments; R=2 already captures most of the benefit in our experiments, and adaptive stopping is a promising direction. Our perturbation parameters were validated through a magnitude sweep (Appendix C), but production systems may face different operational magnitudes than those we modeled. Finally, token-F1 on our open-domain replications is lower than on a fine-tuned QA model (0.07–0.14 across all three corpora, consistent with the Flan-T5-large extractive-QA ceiling); this is a property of the generator rather than any of the retrieval pipelines, and is orthogonal to the stability claim by design (§4.3). Replicating with a task-fine-tuned or larger generator is left for future work.

Ethics Statement

All experiments use publicly available datasets and open-source models; no human subjects were involved and no harmful content was created. All datasets are used under their respective open licenses (SQuAD, TriviaQA, HotpotQA, and NQ-open under CC BY-SA; Wikipedia under CC BY-SA; models under MIT or Apache 2.0 licenses), and our benchmark RAGSTAB will be released under an MIT license. The instability phenomenon could theoretically be exploited by adversaries repeatedly querying a RAG system to discover favorable answers; documenting the vulnerability is preferable to leaving it unknown, and STABRAG provides a defense.

References

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

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. In *Proc. CAIN*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, et al. 2022. Improving language models by retrieving from trillions of tokens. In *Proc. ICML*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proc. AAAI*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for RAG systems. In *Proc. SIGIR*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2024. RAGAS: Automated evaluation of retrieval augmented generation. In *Proc. EACL System Demonstrations*.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proc. NAACL-HLT*.

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

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

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, et al. 2023. Atlas: Few-shot learning with retrieval augmented language models. *JMLR*, 24(251):1–43.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proc. EMNLP*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proc. ACL*.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, et al. 2019. Natural Questions: A benchmark for question answering research. *TACL*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proc. NeurIPS*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *TACL*, 12:157–173.

Yu A. Malkov and Dmitry A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE TPAMI*, 42(4):824–836.

Shintaro Ozaki, Yuta Kato, Siyuan Feng, Masayo Tomita, Kazuki Hayashi, Wataru Hashimoto, Ryoma Obara, Masafumi Oyamada, Katsuhiko Hayashi, Hidetaka Kamigaito, and Taro Watanabe. 2025. Understanding the impact of confidence in retrieval augmented generation: A case study in the medical domain. In *Proc. BioNLP Workshop (ACL)*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proc. ACL*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. EMNLP-IJCNLP*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proc. ACL*.

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

Viju Sudhi, Sinchana Ramakanth Bhat, Max Rudat, and Roman Teucher. 2024. RAG-Ex: A generic framework for explaining retrieval augmented generation. In *Proc. SIGIR*.

Baiqiang Wang, Dongfang Zhao, Nathan R. Tallent, and Luanzheng Guo. 2025. On the reproducibility limitations of RAG systems. *arXiv preprint arXiv:2509.18869*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proc. ICLR*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *Proc. ICLR*.

A Threshold Sensitivity Analysis

Our answer-change metric flags a question as changed if semantic distance ≥ 0.05 or ROUGE-L ≤ 0.85 across any pairwise comparison. Table 11 shows that the change rate varies within a 1.0pp range (7.6% to 8.6%) across the entire 4×3 threshold grid, with ROUGE-L the dominant criterion: relaxing RL from 0.80 to 0.90 raises change rates by up to 1.0pp at every semantic threshold, while tightening semantic distance from 0.03 to 0.10 lowers rates by at most 0.6pp at any RL value. At our chosen operating point (sem. ≥ 0.05, RL ≤ 0.85), perturbing either threshold by one grid step changes the headline number by at most 0.4pp, well within the bootstrap CI for our 8.2% point estimate.

The semantic distance criterion becomes relevant under perturbation conditions, where embedding noise produces more graded divergence. We retain both criteria to handle the full range of instability sources.

Table 11: Threshold sensitivity (ANN, k=10, Flan-T5). Change rates vary within ±0.5pp across the grid; ROUGE-L is the dominant criterion (loosening it from 0.80 to 0.90 raises rates by 0–1.0pp), while semantic distance has a smaller effect (tightening from 0.03 to 0.10 lowers rates by up to 0.6pp).

Sem. dist. $\geq$	RL $\leq$ 0.80	RL $\leq$ 0.85	RL $\leq$ 0.90
0.03	8.2%	8.2%	8.6%
0.05	8.2%	8.2%	8.6%
0.08	7.8%	8.0%	8.6%
0.10	7.6%	7.8%	8.6%

B Simpler Baselines Comparison

Table 12 compares four methods using the same R=8 retrieval runs on Flan-T5. **Single-run baseline**: the answer from a single ANN retrieval (standard RAG). **STABRAG (majority vote)**: Algorithm 1, selecting the most frequent normalized answer across R runs. **STABRAG-Gen (consensus)**: instead of voting over existing answers, this variant constructs a consensus passage set (Stage 1 of Algorithm 1) and generates a single new answer from it. **Exact oracle**: deterministic exact search (zero instability).

At k=10, STABRAG modifies only 12 of 500 answers (2.4%). Among those 12 cases, STABRAG improves F1 in 3 cases and never degrades it (0 worse, 9 ties), confirming that majority vote selectively corrects unstable answers without introducing errors. STABRAG-Gen achieves identical F1, suggesting that the consensus passage set captures the same information as the majority-vote answer.

Table 12: Method comparison (Flan-T5, R=8, ANN).

Method	k=5		k=10	
	F1	EM	F1	EM
Single-run baseline	.052	.012	.062	.018
STABRAG (majority vote)	.051	.012	**.066**	**.022**
STABRAG-Gen (consensus)	.051	.012	**.066**	**.022**
Exact oracle	.059	.020	.074	.022

Both STABRAG variants outperform the single-run baseline at k=10 (0.066 vs. 0.062) while the exact oracle represents the theoretical ceiling (0.074).

C Perturbation Magnitude Sweep

We sweep perturbation magnitude independently for query noise ($\sigma \in \{0.001, 0.005, 0.01, 0.02, 0.05\}$) and score jitter ($\sigma \in \{0.0001, 0.0005, 0.001, 0.005, 0.01\}$), all at k=10 (n=200, Flan-T5). Table 13 reports the results.

Table 13: Perturbation magnitude sweep (k=10, n=200, Flan-T5). Query noise degrades Jaccard proportionally; score jitter maintains Jaccard = 1.000 at every magnitude.

Type	σ	Change	Jaccard	Rank OL
Query noise	0.001	24.0%	0.969	0.973
Query noise	0.005	51.5%	0.879	0.888
Query noise	0.010	60.5%	0.780	0.792
Query noise	0.020	69.5%	0.626	0.639
Query noise	0.050	81.5%	0.303	0.307
Score jitter	0.0001	4.5%	1.000	0.997
Score jitter	0.0005	11.5%	1.000	0.986
Score jitter	0.001	21.5%	1.000	0.971
Score jitter	0.005	40.0%	1.000	0.884
Score jitter	0.010	46.0%	1.000	0.801

Two patterns emerge. First, instability increases with noise magnitude but with diminishing marginal returns: a 50× increase in query noise (0.001 to 0.05) yields a 3.4× increase in change rate (24.0% to 81.5%). This sublinear relationship partly reflects ceiling effects as the metric approaches 100%, but the practical implication remains: even small perturbations (σ=0.001) produce substantial instability (24.0%). Second, score jitter maintains Jaccard = 1.000 at *every* magnitude tested, confirming that jitter operates purely through rank reordering regardless of magnitude. This clean mechanistic separation provides practitioners with a diagnostic: if instability co-occurs with Jaccard = 1.0, the source is rank-level; if Jaccard degrades, the source is embedding-level.

D Qwen2.5-3B: Context-Length Interaction

The main paper (§4.7) reports that Qwen2.5-3B replicates the rank-to-answer coupling direction but exhibits absolute change-rate magnitudes dominated by a non-retrieval pathology. This appendix documents that pathology so readers can interpret Qwen's numbers in Table 6 correctly.

Qwen's token-F1 on SQuAD is near zero (0.001 at k=5, 0.007 at k=10) because the model produces verbose explanatory responses that do not align with SQuAD's short extractive gold answers. Our stability metric is still well-defined – it compares the model's answers to itself across runs, not to gold – but the interpretation of change magnitudes requires care. At low k, change rates are near zero (Table 6: 0.0% at k=5, 2.2% at k=10). At k=20, change rates cliff to 62.2% on ANN and 99.0% on PERT.

The cliff coincides with proportional truncation reducing per-passage context to roughly 47 tokens at k=20 (1024-token budget shared across 20 passages minus prompt overhead), below the length at which Qwen's instruction-following behavior is stable in our setup. We therefore interpret Qwen's k=20 result not as a clean demonstration of the retrieval-instability mechanism but as evidence that retrieval instability can interact with other failure modes in a non-additive way: once per-passage context drops below a model-specific minimum, small perturbations suffice to trigger large output divergences. The rank-to-answer coupling on the regime where Qwen's responses are stable ($k \leq 10$) is ρ=0.29, $p<10^{-31}$, matching Flan-T5's coupling magnitude.

We include Qwen as evidence that the retrieval-instability phenomenon extends to a third generator and a third architecture-organization source, without claiming that its absolute magnitudes at k=20 are measuring only retrieval.

E Extended Analysis and Discussion

This appendix expands on three analysis points from §6.

Ecological validity. A natural objection is that production rerankers already solve the problem. Our reranker results (§5.1) partially support this: reranking eliminates the pure HNSW case. But under realistic operational perturbations (embedder updates, mixed-precision arithmetic), a 45.8% change rate persists with BGE-large plus a cross-encoder. This is not a failure of the reranker, which by design cannot repair a candidate set that perturbation has mutated. Many deployments also omit the reranker for latency or cost reasons, and the pure-HNSW instability reappears there in full. RAGSTAB provides the measurement protocol and STABRAG provides a complementary defense.

Why does rank instability change answers? The passage at position 1 tends to dominate generation; HNSW reordering changes the leader and the answer follows. This "lost in the middle" effect (Liu et al., 2024) appears amplified in decoder-only models: Mistral-7B (ρ=0.50) shows substantially tighter rank-to-answer coupling than Flan-T5 (ρ=0.30). That three independently trained models all show highly significant coupling ($p<10^{-31}$) establishes the phenomenon as a property of the RAG architecture rather than of any one model. Instability also grows with k because (i) more orderings become available to perturb and (ii) proportional truncation (§4.7) reduces each passage to roughly 47 tokens at k=20, degrading per-passage context.

Prompt-level mitigation fails. An alternative to structural mitigation (reranking, consensus) is to simply tell the generator that passage order is unreliable. We test this by shuffling the retrieved passage order and annotating each passage with its retrieval score. At k=10 (ANN, Flan-T5), this degrades F1 from 0.066 to 0.054 relative to the baseline passage ordering, without reducing instability. The generator relies more on positional priors than on explicit relevance signals provided in the prompt. Structural interventions that produce a deterministic passage order (reranking) or aggregate across multiple retrievals (STABRAG) are substantially more reliable than prompt-level hints.

F Extended Experimental Results

This appendix provides supplementary visualizations of the main experimental results.

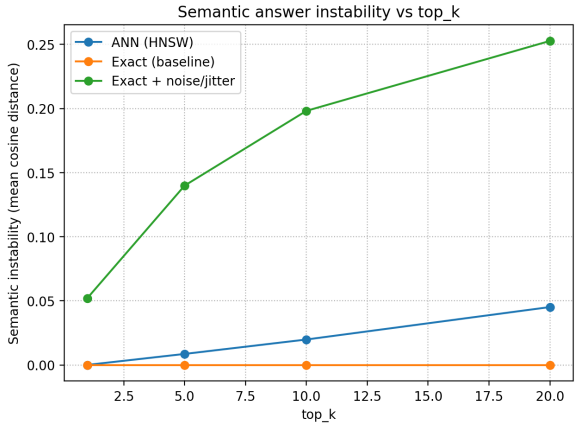

Figure 2: Semantic cosine distance between answer pairs grows with k under both ANN and PERT, while the EXACT control remains at zero (Flan-T5).

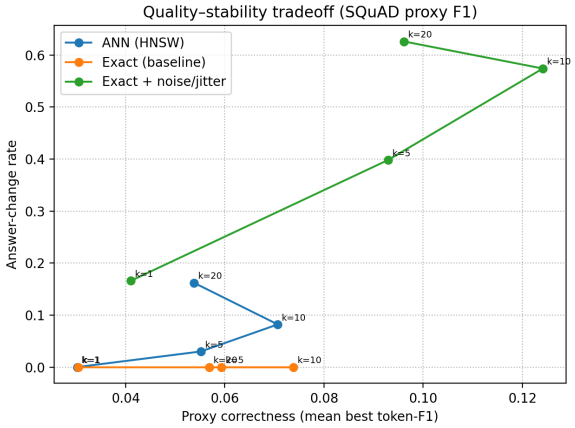

Figure 5: Quality–stability frontier on Flan-T5. Increasing k improves $\mathrm{F1}_{\mathrm{best}}$ but worsens the answer-change rate.

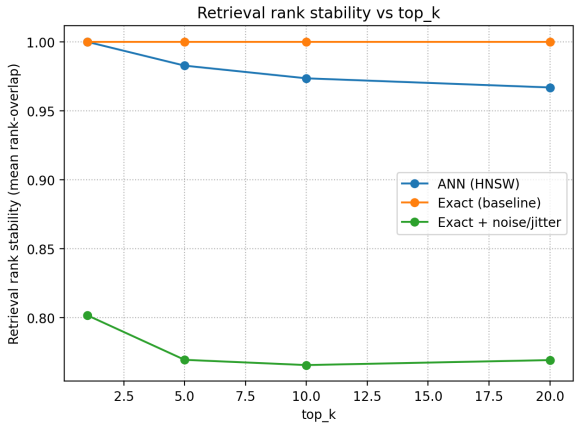

Figure 3: Retrieval rank stability as a function of k. ANN degrades only mildly (Jaccard remains 1.0; rank overlap drops to 0.97), whereas PERT drops sharply because passage sets also change.

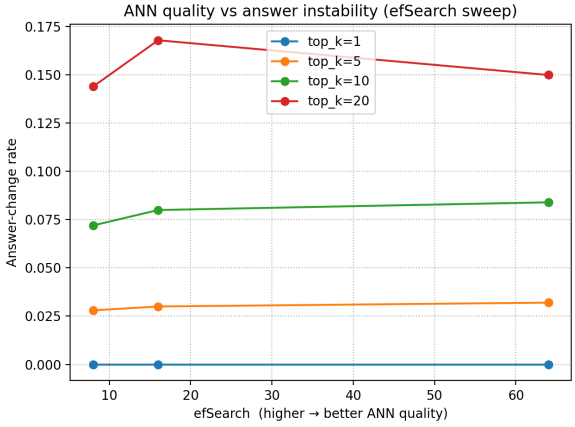

Figure 6: `efSearch` sweep on the ANN variant. Across a 4× range, change rates vary within the 1–2pp run-to-run measurement noise.

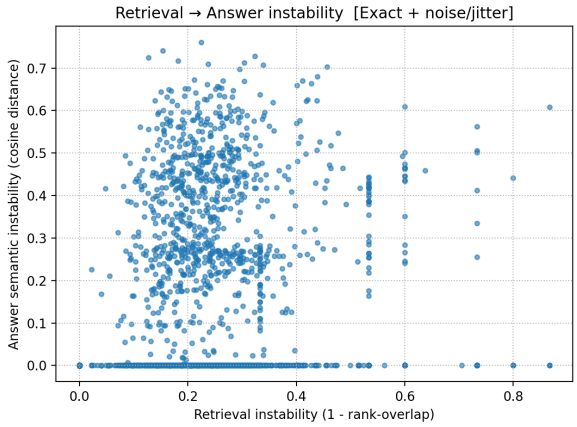

Figure 4: Per-question retrieval instability predicts answer instability (PERT, n=2,000). Questions with larger rank perturbations show larger answer semantic distances.

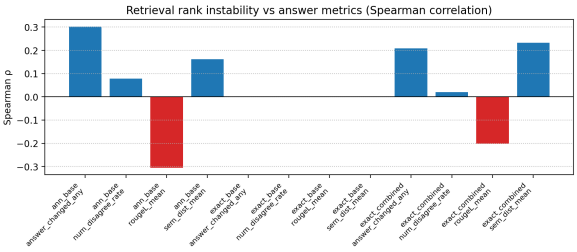

Figure 7: Spearman ρ between rank instability and answer metrics on Flan-T5. Positive bars correspond to the change-rate coupling reported in Table 2; negative bars show the mirror-image effect on ROUGE-L.

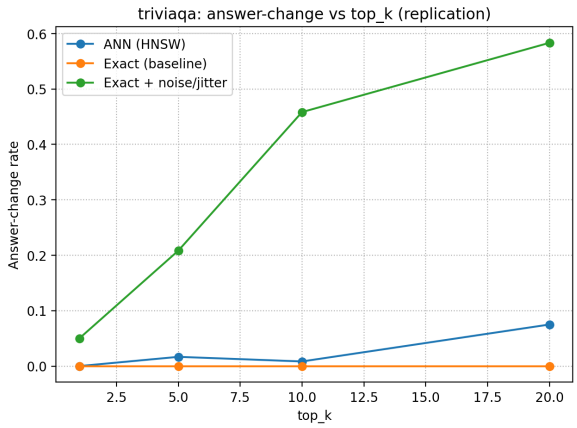

Figure 8: TriviaQA replication (n=120). The ordering of the three variants matches SQuAD, though absolute magnitudes are lower on this smaller corpus.

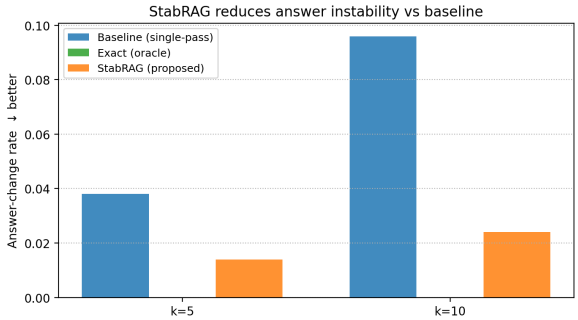

Figure 9: StabRAG reduces the answer-change rate relative to the single-run baseline across both k=5 and k=10 (Flan-T5, R=8).

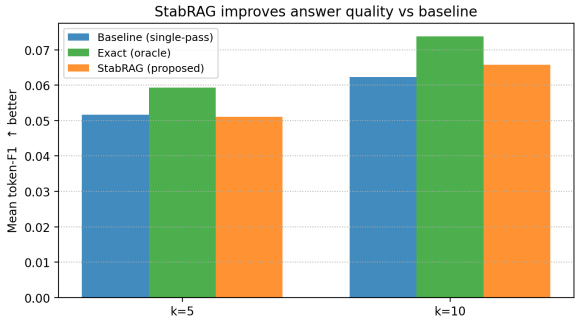

Figure 10: F1 is preserved or slightly improved by StabRAG relative to the single-run baseline (Flan-T5, R=8).

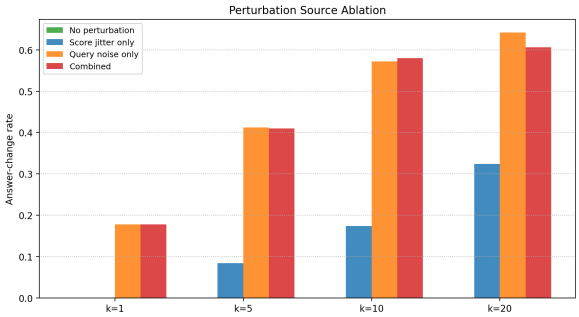

Figure 11: Perturbation-source ablation. Query-embedding noise dominates score jitter by 3.3× at k=10, and the two sources do not compound (combined ≈ noise-only).

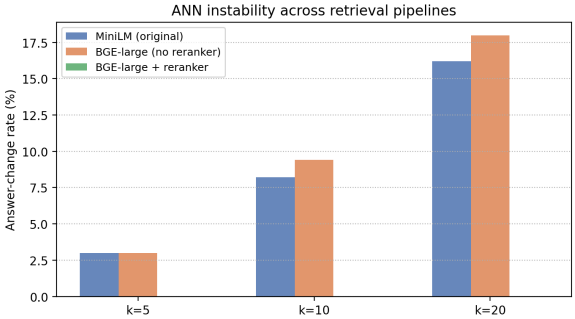

Figure 12: Pipeline comparison on the ANN variant. A stronger embedder (BGE-large) does not reduce instability on its own, but adding a cross-encoder reranker eliminates it at every k.

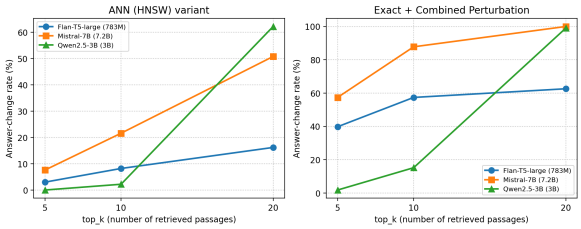

Figure 13: Cross-generator instability profiles (visualization of Table 6). Flan-T5 rises gradually; Mistral-7B is consistently the most unstable; Qwen2.5-3B is stable at low k but cliffs at k=20 due to context truncation rather than retrieval alone (§4.7).