Research Article

In RAG We Trust? Measuring Robustness of Retrieval-Augmented Generation Under Post-Retrieval Context Tampering

Iliano Fasolino[1]

1. Department of Computer Science, University of Milan, Italy

Retrieval-augmented generation grounds a language model in retrieved documents, and that grounding is also an attack surface: an adversary who edits retrieved text, without touching the model or the retriever, can feed the model fluent falsehoods. We measure this on a 4-bit Llama 3.1 8B system with a MiniLM retriever over FEVER evidence. This version replaces the measurement rather than reinterpreting it. The earlier sweep contained an off-by-one that made a zero-corruption cell impossible, so the row published as a clean baseline carried one corrupted passage; a seeded re-run of 1,911 generations supplies a real control, logs how many passages each corruption rule actually modified, and adds closed-book and irrelevant-context arms. Three results follow. Scored on corruption actually applied, the fooled rate runs 0.4%, 35.7%, 54.0% and 72.0% for zero through three corrupted passages, the zero cell being an internal control with no adoption at all. The rules do not deliver the corruption they request: asked for three passages, numeric substitution achieves it in 6% of cells, so the three rules were never comparable and we withdraw the ranking we previously published. Controls reframe the rest, since the generator is more accurate with no context (83.7%) than with clean retrieved context (77.6%), and with irrelevant context it abstains on every question rather than using knowledge it demonstrably has.

Correspondence: papers@team.qeios.com — Qeios will forward to the authors

I. Introduction

Large language models answer fluently but carry two known liabilities: their knowledge is frozen at training time, and they sometimes state false facts with confidence. Retrieval-augmented

 doi.org/10.32388/7HXKQE.3

generation [1] addresses both by fetching documents at query time and placing them in the prompt as evidence, with the instruction to answer from that evidence rather than from memory.

The same design creates a dependency. A RAG answer is only as trustworthy as the text that reaches the prompt. Shi et al. [2] showed that merely irrelevant context pulls a model off track. Corrupted but on-topic context is a stronger lever, because it wears the appearance of the evidence the model was told to trust. Recent work on retrieval corruption shows that a small number of crafted passages can steer a RAG system toward an attacker's answer [3], that text reaching a model through a retrieval or tool channel is an injection surface in deployed systems [4], and that poisoning the corpora behind such systems is practical at scale [5].

What is still thin is a plain measurement on the kind of system a practitioner runs on modest hardware: how far accuracy moves as corrupted context grows, whether the type of corruption matters, and whether a failing model fabricates or refuses. This paper reports such a measurement twice, because the first attempt was instrumented badly enough that the second is a replacement rather than a revision.

Contributions. We rebuild the measurement around a seeded harness that expresses the corruption level as a count of passages, asserts that the two prompts are byte-identical when that count is zero, randomises which arm is generated first, and logs the edits actually applied to each passage. On 1,911 generations this yields: a dose response scored on corruption actually applied, with a genuine zero-dose control; an edit-success rate showing that the three corruption rules deliver very different amounts of corruption for the same nominal level; a direct measure of poison adoption, separated from correctness; and closed-book and irrelevant-context controls establishing what retrieval was worth in the first place. We also state what the earlier analysis got wrong and which of its claims we withdraw.

II. Threat Model

We fix the adversary before describing the system, because the scope of the measurement follows from it.

The adversary can rewrite the text of passages after retrieval and before the prompt is assembled. It cannot retrain or fine-tune the generator, cannot alter the embedding model or the index, and cannot change which passages are retrieved. This is the tampered-context setting: a compromised cache or serving layer between retriever and prompt, a document store edited in place, or a rendering step that rewrites passage text.

We do not model index poisoning, where an attacker inserts documents so that they are retrieved at all. That threat is real [3][5] but it turns on retrieval success, which this study does not manipulate. Claims here concern what a model does with corrupted evidence once that evidence is in front of it.

III. Method

A. Retrieval and Generation

Questions and passages are embedded with all-MiniLM-L6-v2 [6] into 384-dimensional vectors, normalised to unit length and stored in a FAISS flat inner-product index [7], so the inner product equals cosine similarity:

$$\mathrm{sim}(q, d) = \frac{\mathbf{e}_q \cdot \mathbf{e}_d}{\|\mathbf{e}_q\| \, \|\mathbf{e}_d\|}. \tag{1}$$

The index is scanned exhaustively. The top $k = 3$ passages go to the generator, Llama 3.1 8B Instruct quantized to 4 bits (GGUF Q4_K_M) under llama.cpp on CPU, which fits the 8 GB machine used throughout. Decoding uses temperature 0.1 with an explicit per-generation seed. The prompt, reproduced in Section VII, instructs the model to answer only from the supplied context and to reply "Not enough information" otherwise.

B. Corruption Rules and Dose

A corruption rewrites a retrieved passage so that it asserts something false while staying fluent and on topic. *Entity swap* replaces named entities with same-type alternatives, turning "the Eiffel Tower is located in Paris, France" into "... in Madrid, Spain". *Number swap* perturbs a numeric value, turning "over 13,000 miles" into "over 3,000 miles". *Negation* inserts a negation, so "Barack Obama was the 44th President" becomes "... was never the 44th President". These are rule-based edits over a fixed substitution table, not model-generated adversarial text.

The corruption level is the *number of passages to corrupt*, $m \in \{0, 1, 2, 3\}$, passed to the routine as a count. The earlier sweep passed a ratio p and converted it with $m = \max(1, \lfloor kp \rfloor)$, a formula that never returns zero and maps both $p = 0$ and $p = 1/3$ to $m = 1$; Section V-A reports the damage.

A requested count is not a delivered one. A rule can only corrupt a passage in which it finds something to change, and many FEVER evidence sentences contain no swappable numeral or no entity in the

substitution table. The harness therefore records, per passage, whether an edit was applied, and we report the resulting *applied dose* alongside the requested one.

C. Outcome Labels

Each run pairs one clean answer with one corrupted answer over the same retrieved list, under seeds fixed by cell coordinates. Four labels are recorded.

Correctness. The original criterion was presence of a reference keyword. That criterion is not neutral across rules: negation leaves the keyword in place, entity swap removes it by construction, and an answer that denies the fact (“Madrid, not Paris”) still contains the keyword. We use a *polarity-aware* criterion that resolves abstention first, reads the yes/no stance for boolean questions, and requires the keyword to fall outside the scope of a negation cue.

Fooled rate. Runs where the clean answer was correct and the corrupted answer was not, divided by the runs that were correct before the attack. The earlier version divided by all runs while describing it as the former.

Abstention. Answers that decline to commit, matched against the fixed phrase list in Section VII. Abstention and fooled are not disjoint, so stacked figures use a resolved three-way outcome on clean-correct runs only.

Poison adoption. Whether the answer contains a value introduced by the corruption, tested against the edits actually logged for that run. Adoption is a different axis from correctness: an answer can abstain and still quote the corrupted value. Earlier versions used a lexical-overlap proxy instead,

$$\text{lowovl} = \mathbf{1}\left[\frac{|W_a \cap W_d|}{|W_a|} < 0.3\right], \tag{2}$$

which has two defects we quantify in Section V-H: W_d was the arm’s own, hence corrupted, passages, so an answer that copied the poison scored as well supported; and abstentions were excluded before evaluation and counted as supported, so the measure fell mechanically as abstention rose.

IV. Experimental Design

The evaluation set holds 49 indexed query slots drawn from 37 distinct question texts. The 12 repeats exist because the set was assembled by listing questions per rule and then appending a mixed block that

re-used earlier questions. They are not independent observations, and every interval in this paper resamples the 37 distinct texts.

Ten information-rich passages covering the queried facts were written by the author and added to the index, because raw FEVER evidence sentences are often too terse to support a full answer. Retrieval diagnostics were run against that injected map (script `retrieval_diag.py`; reconstructed clean contexts shipped as `clean_contexts.json`), confirming that the top-k set usually includes a topic-relevant injected passage and that a substantial share of retrieved text is author-written rather than raw FEVER evidence. Hard recall@3 percentages are not restated here because the printed summary was not reproducible from the shipped package alone. Qualitative reading is unchanged: what follows is not mainly retrieval failure. The misses cluster on one topic, and one query, "How many escalators does Burj Khalifa have?", has no supporting passage in the index at all.

Two sweeps are referenced. The *published sweep* crossed 49 slots with three rules and four nominal levels for 588 runs; its analysis is corrected in Sections V-A and V-H. The *seeded sweep* supplies the results of this paper: three rules by three requested doses by 49 queries by three independent corruption draws, 1,323 grid cells, plus 147 runs each of a clean arm, a zero-dose control, a closed-book arm and an irrelevant-context arm, for 1,911 generations and about 38 hours of CPU.

V. Results

A. The Published Sweep Had No Zero Dose

Under $m = \max(1, \lfloor kp \rfloor)$ with $k = 3$, the four nominal levels realise one, one, two and three corrupted passages. The published no-attack baseline was an attacked cell and the first two levels were a single dose.

The data carry the signature. At nominal $0/3$ the corrupted arm quotes content that exists only in corrupted text: "The context states the Eiffel Tower is located in Madrid, Germany"; "according to statement 2, the Eiffel Tower is located in Madrid, Spain, not France". Correctness at that cell is 115 clean against 102 corrupted, with discordant counts $b = 14$, $c = 1$ and exact McNemar $p = 9.8 \times 10^{-4}$. A reviewer of the previous version judged that asymmetry too large for symmetric decoding noise and asked for an artifact search. This is the artifact.

B. What a Real Zero Dose Looks Like

The seeded harness makes the zero cell real: it asserts the two prompts are byte-identical before generating and randomises which arm runs first. Across 147 control runs the assertion holds in 100% of cases and no passage is edited.

The two generations then agree verbatim in 88.4% of runs, so decoding at temperature 0.1 is not deterministic. What matters is that the disagreement is symmetric: correctness splits 115 against 114, with discordant counts $b = 1$, $c = 0$ and exact McNemar $p = 1$. Which arm is generated first does not matter either, at 89.7% agreement against 87.0%. Decoding noise of this shape cannot manufacture the $b = 14, c = 1$ asymmetry of Section V-A, because it has no direction. The corruption does.

C. The Rules Do Not Deliver the Dose They Request

	1 requested		2 requested		3 requested	
Rule	**mean**	**exact**	**mean**	**exact**	**mean**	**exact**
Entity swap	0.52	52%	1.06	40%	1.59	29%
Number swap	0.41	41%	0.91	18%	1.31	**6%**
Negation	0.57	57%	1.26	44%	1.92	37%

Table I. Passages actually corrupted for each requested count, seeded sweep. A rule can only edit a passage in which it finds a target.

Table I and Figure 1 report the gap. Asked to corrupt all three passages, numeric substitution succeeds in 6% of cells, because most FEVER evidence sentences carry no swappable numeral. Negation, which only needs a copula, manages 37%.

Two consequences follow, and both are consequential for the earlier analysis. Any dose response indexed on the requested level is attenuated by an unknown and rule-dependent amount. And the three rules were never delivering comparable corruption, so comparing their strength at equal nominal dose compares different attacks.

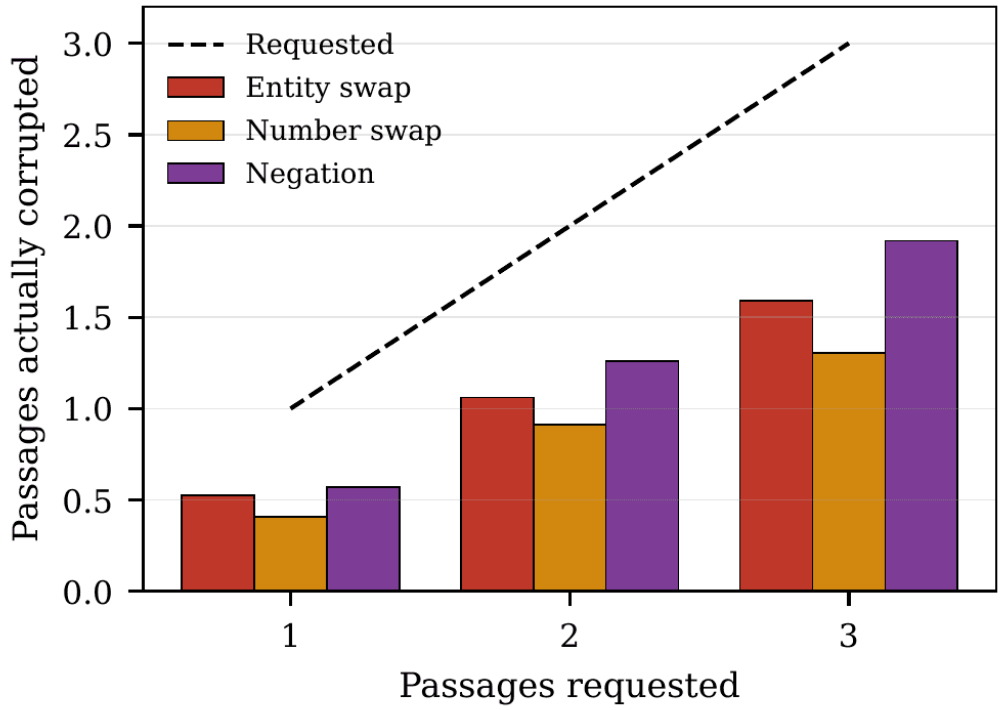


Figure 1. Requested against applied corruption. The dashed line is what was asked for. Numeric substitution falls furthest short, and the shortfall grows with the requested dose.

D. Dose Response on Corruption Actually Applied

Applied	n	Fooled	95% CI	Adoption	Abstain
0	418	0.4	$[0.0,\ 1.3]$	0.0	31.3
1	511	35.7	$[25.0,\ 47.6]$	10.8	34.8
2	289	54.0	$[40.1,\ 66.5]$	23.5	46.0
3	105	72.0	$[51.5,\ 92.0]$	15.2	62.9

Table II. Seeded grid scored on the dose actually applied. The zero row is an internal control: a corruption was requested but nothing could be modified, so the model saw clean context under otherwise identical conditions.

doi.org/10.32388/7HXKQE.3

Table II is the central result. On a clean-correct base the fooled rate rises from 0.4% with nothing modified to 72.0% with three passages modified, and the interval at zero excludes every other row.

The zero row deserves emphasis because the published sweep had no equivalent. These are 418 cells in which the harness requested a corruption, the rule found nothing to edit, and the model therefore saw clean context while every other condition was held fixed. The fooled rate there is 0.4% and adoption is exactly zero. That is the behaviour a no-attack cell should show, and it is what the published $0/3$ row failed to show.

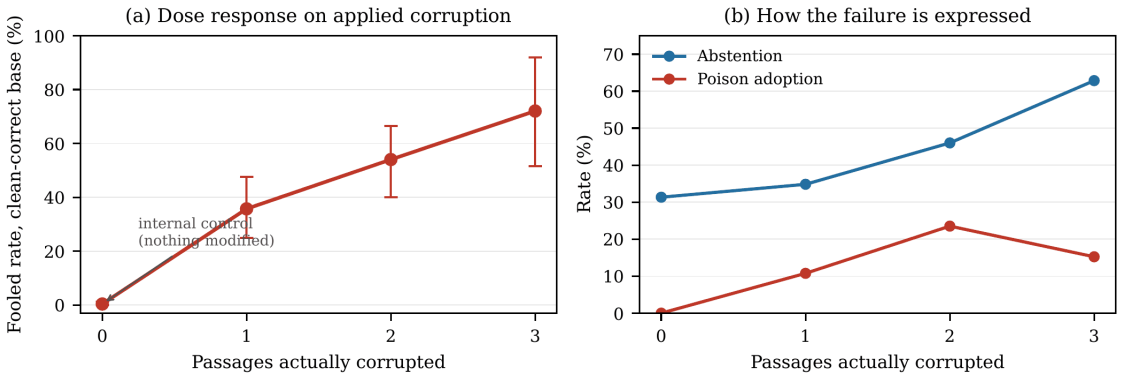


Figure 2. Panel (a): fooled rate against passages actually corrupted, with query-level bootstrap intervals. The leftmost point is the internal control. Panel (b): abstention and poison adoption over the same axis. Abstention rises throughout; adoption rises then falls as refusal takes over at full corruption.

Figure 2 adds how the failure is expressed. Abstention climbs monotonically, from 31.3% to 62.9%. Adoption rises from zero to 23.5% at two corrupted passages and then falls back to 15.2% at three, which is consistent with refusal displacing commitment once the context is wholly corrupted: the model has less and less that it is willing to assert, including the poison.

E. The Rules Are Still Not Separable

Rule	Fooled (%)	95% CI
Entity swap	40.1	$[28.3,\ 51.1]$
Number swap	37.4	$[24.6,\ 50.5]$
Negation	26.6	$[15.0,\ 39.8]$
Entity $-$ Number	$+1.1$	$[-11.0,\ +12.5]$
Entity $-$ Negation	$+7.5$	$[-0.5,\ +16.3]$
Number $-$ Negation	$+6.4$	$[-7.1,\ +19.9]$

Table III. Fooled rate by rule over three corruption draws, and pairwise differences paired by query text. No interval excludes zero.

The previous version reported that entity swap flips the most answers and negation is the most resilient. Table III does not support it, now on three independent corruption draws rather than one. Point estimates preserve the order, every pairwise interval contains zero, and the closest contrast, entity against negation, has a lower bound of -0.5 points.

The comparison is in fact worse posed than that. Table I shows the rules deliver different amounts of corruption, so a fair test would hold applied dose equal. Restricting to cells where a rule landed all three edits leaves no query answered that way by both entity and number substitution, and only five shared between entity swap and negation. At this corpus the rules cannot be compared at equal delivered corruption, because they cannot deliver equal corruption. We withdraw the ranking and do not replace it.

F. Adoption Separates the Rules Where Fooled Rate Cannot

Rule	Overall	By applied dose		
		1	2	3
Entity swap	15.2%	16%	38%	14%
Number swap	15.2%	16%	32%	100%
Negation	**1.1%**	1%	3%	2%

Table IV. Poison adoption in the seeded sweep, tested against the edits actually applied.

Where the fooled rate cannot tell the rules apart, adoption can. Table IV and Figure 3(b) show entity and numeric substitution adopted at 15.2% each, against 1.1% for negation. The model repeats a substituted place or figure readily and almost never repeats an inserted negation. This is the one behavioural difference between the rules that this study establishes, and it is invisible to a correctness-based comparison.

Two further observations bear on how the earlier metrics were read. Adoption and abstention are not exclusive: 57.6% of adopting answers also decline to commit, quoting the corrupted value and then reporting the context insufficient. Refusal is therefore not evidence that corruption was rejected. And in the seeded sweep no adopting answer was scored correct, so adoption is cleanly distinguishable from the fooled rate rather than a relabelling of it.

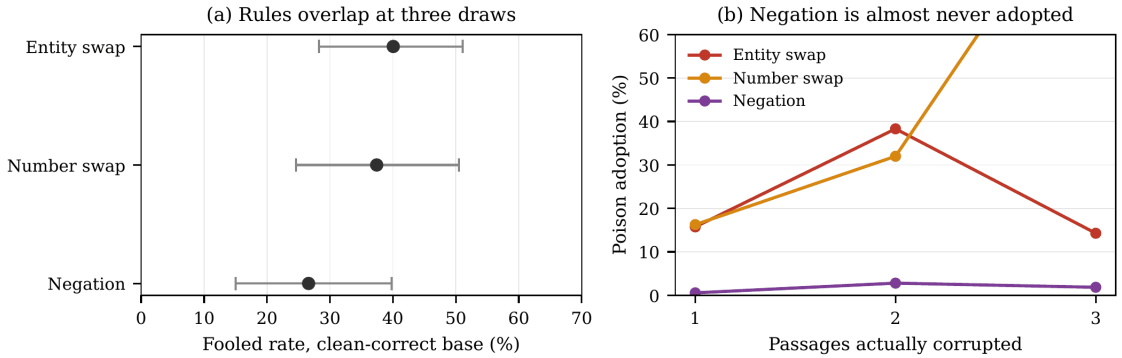


Figure 3. Panel (a): fooled rate by rule with bootstrap intervals; the three overlap. Panel (b): poison adoption over applied dose, where negation separates sharply from the other two.

G. What the Context Was Worth

Arm	Correct (%)	Abstain (%)
Clean context	77.6	12.2
Closed book, no context	**83.7**	0.0
Irrelevant context	0.0	100.0

Table V. Controls, 147 runs each. The closed-book arm receives no context and is told to answer from its own knowledge; the irrelevant arm receives three true but off-topic passages under the standard instruction.

Neither earlier version established what retrieval contributed. Table V and Figure 4 settle it, and the answer is uncomfortable: the generator is more accurate with no context at all than with the clean retrieved context, by about six points. Eight questions are answered correctly closed-book and lost under clean retrieval, and they are the questions where retrieval misses. “What team did Kaepernick play for?” returns “San Francisco 49ers” without context and “Not enough information” with it.

The irrelevant arm gives the mechanism. Given three true but off-topic passages the model answers correctly zero times out of 147 and abstains in every single run. It never falls back on knowledge it

demonstrably has. The instruction to answer only from context is obeyed absolutely, so context does not compete with parametric knowledge, it replaces it.

That reframes the dose response. Corruption is not overriding what the model believes; it is the only thing the model is permitted to believe. The degradation in Table II is the cost of binding an instruction following model to evidence, realised when the evidence is corrupted. It also bounds external validity: these questions concern well-known entities the generator already knows, and a corpus of facts outside its training data would behave differently in ways this design cannot predict.

One caveat on comparability. The closed-book arm necessarily uses a different system instruction, since “answer only from the context” is undefined with no context, so 83.7% is not a strictly matched condition. The direction does not depend on the margin.

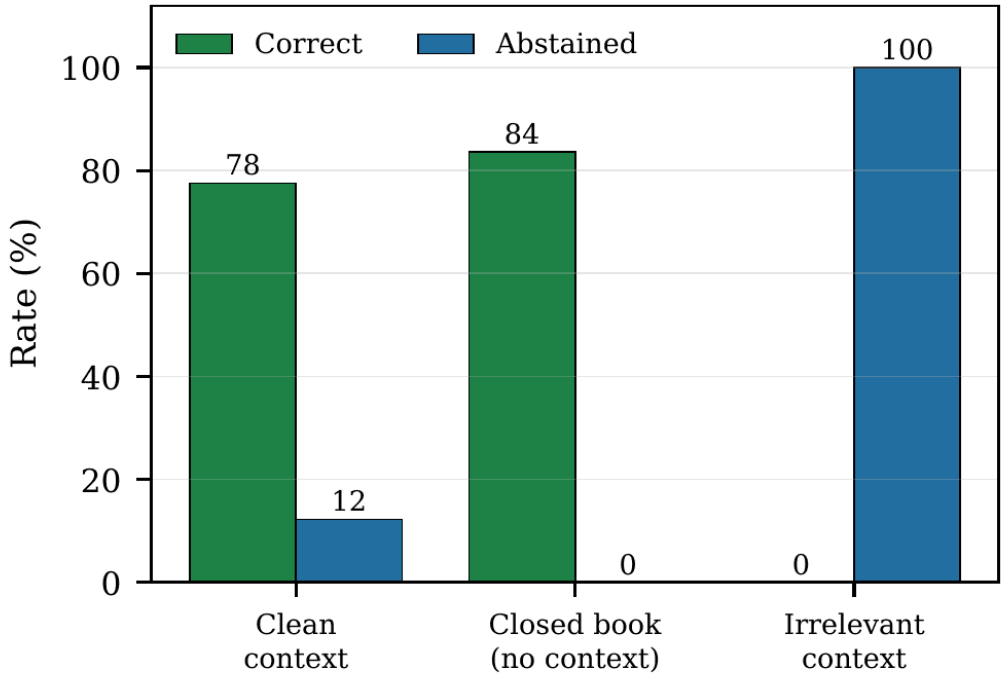


Figure 4. Controls. Removing the context raises accuracy; replacing it with true but irrelevant text collapses the model to universal abstention rather than to its own knowledge.

H. Claims Withdrawn from the Published Analysis

Four statements from the previous version do not survive and are withdrawn here rather than quietly dropped.

doi.org/10.32388/7HXKQE.3

The zero-poison baseline. It was not one, as Section V-A shows, and its residual was attributed to decoding noise and a strict matcher. Decoding noise exists but is symmetric, so it cannot account for that cell.

The headline drop. Reporting 77.9% on the clean arm against 43.5% at full corruption mixes a between-arm gap with the attack. On the published table the nominal within-arm attack-only drop is 69.4% to 43.5%, or 25.9 points; the rebinned applied-dose drop from realised dose 1 (66.7%) to dose 3 (43.5%) is 23.2 points. In the seeded sweep the corresponding contrast is the zero-to-three row of Table II.

The strategy ranking. Withdrawn in Table III, and further undermined by Table I, which shows the rules were not delivering comparable corruption in the first place.

The fall in unsupported generation. Eq. (2) scores every abstention as supported, and abstention is what rises under attack, so the measure declines arithmetically. On the published table the all-answers rate fell from 28.9% to 12.2% while the rate over committed answers moved only from 33.1% to 25.4% and was not monotone in between. The proxy is also blind to adoption by construction, since it compares an answer against the corrupted text it received: “located in Madrid, Spain, not France” overlaps the true context at 0.67, far above the threshold. Direct adoption measurement replaces it.

A fifth claim is weakened rather than withdrawn. Abstention was offered as a cheap warning signal. It still rises with dose, but 57.6% of adopting answers abstain, and a merely irrelevant context produces 100% abstention, so the signal does not distinguish corruption from useless retrieval and does not imply the corruption was rejected.

VI. Related Work

Retrieval-augmented generation was introduced to ground generation in retrievable evidence [1], extending retrieval-integrated pre-training [8]. The failure mode here is the inverse of the intended one: grounding helps on a clean corpus and hurts on a corrupted one.

On the adversarial side, Zou et al. [3] show that injecting a small number of crafted passages into a knowledge base steers a RAG system to attacker-chosen answers, and frame retrieval corruption as an attack surface distinct from prompt manipulation. Greshake et al. [4] show that any channel delivering text into a model’s context functions as an indirect injection vector in deployed applications. Carlini et al. [5] show that poisoning the web-scale corpora behind such systems is practical. Our setting is narrower than all three: the adversary edits already-retrieved text, which isolates the generation stage from retrieval success.

Shi et al. [2] establish that irrelevant context degrades reasoning. Our irrelevant-context arm sharpens that for this setting: under an answer-only-from-context instruction, irrelevant context does not degrade the answer, it suppresses it entirely. Zhou et al. [9] caution that evaluation data can leak into training; we keep the query set fixed across all cells, though we cannot rule out that FEVER-derived facts about well-known entities are present in the generator's training data, and Section V-G suggests they are.

VII. Limitations and Reproducibility

A. What This Study Does Not Establish

The measurement is one quantized 8B model, one retriever, $k = 3$, 37 distinct questions and rule-based corruption. Absolute rates should not be read as general. No pairwise contrast between corruption rules is separable by fooled rate, and at this corpus the rules cannot be equalised on delivered corruption, so we make no ranking claim.

Three gaps remain. Correctness and adoption are both automated and neither was validated against human judgement or an entailment model on a stratified sample; the adoption rule is sharp for substituted entities and numbers and weak for negation, where it looks for an explicit cue rather than a flipped truth value, so 1.1% is a lower bound for that rule. The corruption is rule-based, and an attacker using a model to write internally consistent passages would remove the contradiction cues our entity swaps sometimes leave. And k is fixed at 3 throughout, so this study provides no evidence about whether retrieving more passages helps or hurts; the previous version's advice to raise k alongside source diversity was unsupported and is withdrawn.

B. Reproducibility Details

The system prompt is: *"You are a fact-checking assistant. Answer the question based ONLY on the provided context. If the context doesn't contain enough information, say 'Not enough information'."* The user turn is the numbered context, then `Question:`, then `Answer:`, wrapped in the Llama 3.1 chat template. Abstention is matched, case-insensitively, against: *not enough information*, *cannot determine*, *not provided*, *unclear*, *no information*, *don't know*, *cannot answer*.

Versions are pinned as follows: llama-cpp-python 0.3.16, sentence-transformers 5.2.2, faiss-cpu 1.13.2, numpy 1.26.2, pandas 2.1.4; generator Llama 3.1 8B Instruct GGUF Q4_K_M with SHA-256 `7b064f5842bf9532c91456deda288a1b672397a54fa729aa665952863033557c`; embedder all-MiniLM-

L6-v2; corpus the FEVER dev split [10], 19,597 deduplicated evidence sentences, plus ten author-written passages; hardware an Apple M1 with 8 GB, CPU inference only (`n_gpu_layers=0`). Seeds are derived per generation from a checksum of the cell coordinates. Bootstrap intervals use 10,000 resamples of the 37 distinct query texts with seed 42.

One quirk applies to every run reported here. The prompt is built with an explicit `<|begin_of_text|>` token and the runtime prepends its own, so each prompt carries a duplicated leading token. It is a deviation from the reference chat template, kept in the seeded harness so old and new runs stay comparable, and flagged rather than silently changed.

The artifact contains the corruption code, the seeded harness, the 49 queries with reference keywords, the ten injected passages, the index build script, both run tables, the scoring and analysis scripts and the figure code. Release is pending and will be linked from the version of record.

VIII. Conclusion

We measured a small quantized RAG system under post-retrieval context tampering, replacing an earlier measurement whose instrumentation was faulty. Scored on corruption actually applied, the fooled rate runs 0.4%, 35.7%, 54.0% and 72.0% across zero to three corrupted passages, with the zero cell an internal control that shows no adoption at all. Abstention rises throughout. Adoption peaks in the middle of the range and falls at full corruption as refusal displaces commitment.

The comparative claim we previously published does not survive. The three corruption rules cannot be separated by fooled rate on three independent draws, and they cannot be equalised on delivered corruption either, since numeric substitution lands its full requested dose in 6% of cells. Where they do differ is adoption: substituted entities and figures are repeated at 15.2%, inserted negations at 1.1%.

The practical reading is narrower than before and, we think, more useful. Under an instruction to answer only from retrieved text, that instruction is obeyed absolutely: with off-topic context the model abstained on all 147 runs rather than using knowledge it plainly had, and with no context at all it scored higher than with clean retrieval. Evidence quality therefore sets a hard ceiling, corrupted evidence has no internal competitor, and retrieval should be justified per query set rather than assumed, because on facts a model already knows it can subtract accuracy instead of adding it. Abstention is a weaker signal than we previously suggested, since it accompanies adoption more often than not and is also what useless retrieval produces.

What this design still cannot answer is how any of it behaves on facts outside the generator's training data, which is precisely where retrieval should earn its place, and whether model-written corruption that removes the internal inconsistencies of rule-based edits would be adopted more readily. Both are the next measurements rather than the next argument.

References

1. ^a, ^b *Lewis P, et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." Advances in Neural Information Processing Systems. **33**:9459–9474.*
2. ^a, ^b *Shi F, et al. (2023). "Large Language Models Can Be Easily Distracted by Irrelevant Context." Proc. Int. Conf. Machine Learning (ICML).*
3. ^a, ^b, ^c *Zou W, Geng R, Wang B, Jia J (2025). "PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models." 34th USENIX Security Symposium (USENIX Security 25). 3827–3844.*
4. ^a, ^b *Greshake K, Abdelnabi S, Mishra S, Endres C, Holz T, Fritz M (2023). "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection." Proc. ACM Workshop on Artificial Intelligence and Security (AISec).*
5. ^a, ^b, ^c *Carlini N, et al. (2023). "Poisoning Web-Scale Training Datasets Is Practical." arXiv preprint arXiv:2302.10149.*
6. ^ *Reimers N, Gurevych I (2019). "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks." Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP). 3982–3992.*
7. ^ *Johnson J, Douze M, Jégou H (2019). "Billion-Scale Similarity Search with GPUs." IEEE Trans Big Data. **7**(3):535–547.*
8. ^ *Guu K, Lee K, Tung Z, Pasupat P, Chang M (2020). "REALM: Retrieval-Augmented Language Model Pre-Training." Proc. Int. Conf. Machine Learning (ICML). 3929–3938.*
9. ^ *Zhou K, et al. (2023). "Don't Make Your LLM an Evaluation Benchmark Cheater." arXiv preprint arXiv:2311.01964.*
10. ^ *Thorne J, Vlachos A, Christodoulopoulos C, Mittal A (2018). "FEVER: A Large-Scale Dataset for Fact Extraction and Verification." Proc. Conf. North American Chapter of the Association for Computational Linguistics (NAACL-HLT). 809–819.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.