

Entropy-Triggered RAG: Optimizing Retrieval Efficiency via Token-Level Shannon Entropy

Stanford CS224N Custom Project

Yucheng Yao

Department of Computer Science
Stanford University
ycx@stanford.edu

Abstract

Retrieval-Augmented Generation (RAG) improves language model accuracy on knowledge-intensive tasks but incurs significant latency from unnecessary retrieval calls. We propose Entropy-RAG, which replaces FLARE’s min-probability trigger with Shannon entropy $H(P_t) > \tau$ computed over the full next-token distribution, providing a more principled measure of model uncertainty. We further introduce **Entropy-Weighted Querying (EWQ)**, an original contribution that constructs retrieval queries from tokens with the highest entropy-weighted TF-IDF scores, eliminating FLARE’s costly lookahead forward pass. Experiments on a 150-question subset of StrategyQA using Llama-3-8B with NF4 quantization show that all RAG methods substantially outperform the parametric baseline (+4–5% EM). EWQ achieves competitive accuracy (55.3% EM, within 0.7% of FLARE) while reducing retrieval calls by 21% compared to standard Entropy-RAG, validating that the same uncertainty signal triggering retrieval can also guide what to retrieve.

- Mentor: Minsik Oh
- External Collaborators: None
- Sharing project: No
- AI tools used: Claude (code debugging, report drafting)
- All implementations are original based on cited papers; no external code was reused.

1 Introduction

Large language models (LLMs) achieve strong performance on many NLP tasks but remain prone to hallucination when their parametric knowledge is insufficient [1]. Retrieval-Augmented Generation (RAG) addresses this by grounding generation in retrieved external passages [2], but naïve approaches that retrieve at every step waste computation and can introduce irrelevant context that degrades output quality [3].

FLARE [4] introduced active retrieval, triggering retrieval only when the model exhibits uncertainty. Specifically, FLARE operates at the *sentence level*: after generating a complete sentence, it checks whether any token in that sentence had probability below threshold θ . If so, it performs a lookahead forward pass to generate a draft continuation, masks low-confidence tokens (below β), and uses the remaining tokens as a retrieval query. While effective, this design has two limitations: (1) the sentence-level min-probability check conflates token-level uncertainty with sentence-level decisions, and the lookahead mechanism—not the min-probability signal alone—is responsible for much of FLARE’s retrieval precision; (2) the lookahead forward pass doubles inference cost per retrieval event.

We propose two contributions. First, **Entropy-RAG** replaces min-probability with Shannon entropy $H(P_t) = -\sum_i p_i \log p_i$ computed over the full next-token distribution. Entropy aggregates

information across all vocabulary items rather than relying on a single token’s probability, providing a more robust and theoretically grounded uncertainty measure [5]. Second, **Entropy-Weighted Querying (EWQ)** constructs retrieval queries by selecting tokens with the highest entropy $\times$ TF-IDF scores from the generation so far, entirely eliminating the lookahead pass. Our central hypothesis is that the same uncertainty signal that triggers retrieval can also guide what to retrieve.

We evaluate five retrieval strategies—Parametric (no retrieval), Fixed-Interval, FLARE, Entropy-RAG, and Entropy-RAG+EWQ—on StrategyQA [6], a multi-hop binary reasoning benchmark, using Llama-3-8B-Instruct [7] with NF4 quantization. Our results show that EWQ reduces retrieval calls by 21% versus standard Entropy-RAG while maintaining identical accuracy, validating the dual use of entropy for both triggering and query construction.

2 Related Work

Retrieval-Augmented Generation. RAG [2] augments LLMs with a non-parametric retrieval component, improving factual accuracy on knowledge-intensive tasks. DPR [8] introduced dense passage retrieval using dual encoders, while subsequent work explored tighter integration between retrieval and generation [9].

Active Retrieval. Rather than retrieving at every step, active methods decide *when* to retrieve. FLARE [4] combines a sentence-level min-probability check with a lookahead masking strategy to form retrieval queries. Self-RAG [10] trains the model to emit special retrieval tokens. Our work follows the FLARE paradigm but replaces its uncertainty signal with entropy and its lookahead trigger, arguing it better captures distributional uncertainty than single-token probability.

Uncertainty Estimation in LLMs. Token-level entropy has been used to measure model confidence in calibration studies [1] and selective prediction. Predictive entropy over full sequences has been explored for hallucination detection [11]. We apply token-level entropy specifically as a retrieval trigger, arguing it better captures distributional uncertainty than single-token probability.

Multi-Hop Reasoning. StrategyQA [6] requires implicit multi-hop reasoning, making it a strong testbed for retrieval-augmented methods. Chain-of-thought prompting [12] enables LLMs to decompose such questions into reasoning steps, which we leverage in our 6-shot prompt design.

3 Approach

3.1 Problem Formulation

Given a question q , an LLM generates tokens autoregressively: $y_t \sim P_\theta(\cdot|q, y_{<t}, C)$, where C is the (possibly empty) set of retrieved passages. The challenge is deciding (1) when to populate C via retrieval, and (2) what query to send to the retriever.

3.2 Entropy-Triggered Retrieval

At each generation step t , we compute Shannon entropy over the next-token distribution:

$$H(P_t) = - \sum_{i=1}^{|V|} p_i \log p_i, \quad p_i = P_\theta(v_i|q, y_{<t}, C) \quad (1)$$

Retrieval is triggered when $H(P_t) > \tau$, where τ is a threshold hyperparameter. Unlike FLARE’s min-probability, which examines only $\min_i p_i$ and is trivially small for large vocabularies ($|V| = 128,256$ for Llama-3), entropy captures the *spread* of the distribution. A peaked distribution (confident model) yields low entropy regardless of vocabulary size, while a flat distribution (uncertain model) yields high entropy.

3.3 Entropy-Weighted Querying (EWQ)

When retrieval is triggered, EWQ constructs a query without any additional forward pass. We score each previously generated token y_j ($j < t$) by:

$$\text{score}(y_j) = H(P_j) \times \text{TF-IDF}(y_j, y_{<t}) \quad (2)$$

Algorithm 1 Entropy-RAG with Entropy-Weighted Querying

Require: Question q , threshold τ , top- k , retriever $\mathcal{R}$, LLM P_θ

```
1: Initialize context  $C \leftarrow \emptyset$ , output  $Y \leftarrow []$ , call count  $n \leftarrow 0$ 
2: entropies  $\leftarrow []$  {Store per-token entropy}
3: for  $t = 1$  to  $T_{\max}$  do
4:   Compute logits  $\ell_t = P_\theta(\cdot|q, y_{<t}, C)$ 
5:    $H(P_t) \leftarrow -\sum_i \text{softmax}(\ell_t)_i \log \text{softmax}(\ell_t)_i$ 
6:   Append  $H(P_t)$  to entropies
7:   if  $H(P_t) > \tau$  and  $n < n_{\max}$  then
8:     {Build query from highest entropy  $\times$  TF-IDF tokens}
9:     for each  $y_j \in Y$  do
10:       $s_j \leftarrow \text{entropies}[j] \times \text{TF-IDF}(y_j, Y)$ 
11:     end for
12:      $Q \leftarrow \text{concat}(\text{top-}k \text{ tokens by } s_j)$ 
13:      $C \leftarrow C \cup \mathcal{R}(Q)$ ;  $n \leftarrow n + 1$ 
14:   end if
15:   Sample  $y_t \sim P_\theta(\cdot|q, y_{<t}, C)$ ; append  $y_t$  to  $Y$ 
16:   if  $y_t = \text{EOS}$  then
17:     break
18:   end if
19: end for
20: return  $Y$ 
```

where $H(P_j)$ is the entropy at the step when y_j was generated, and TF-IDF measures the token’s specificity. The top- k tokens by this score are concatenated to form the retrieval query. The intuition is that tokens generated under high uncertainty and with high specificity are precisely the knowledge gaps that retrieval should address.

Algorithm 1 formalizes the complete Entropy-RAG+EWQ generation procedure.

3.4 Baseline Methods

We compare against four baselines: **Parametric-Only**: No retrieval; the LLM generates from parametric knowledge alone. **Fixed-Interval**: Retrieve every $N = 3$ sentences, appending the top- k passages to the context. **FLARE** [4]: Sentence-level active retrieval with lookahead masking. After generating a sentence, if any token had $p_i < \theta$, FLARE generates a lookahead draft, masks tokens with $p_i < \beta$, and uses the remaining tokens as query. **Entropy-RAG** (without EWQ): Use entropy triggering but standard last-sentence query construction (same as Fixed-Interval’s query strategy).

4 Experiments

4.1 Data

We evaluate on StrategyQA [6], a binary (yes/no) multi-hop reasoning benchmark requiring implicit decomposition of questions into reasoning steps. We use subsets from the test split of the wics/strategy-qa dataset on HuggingFace: 150 questions for the main comparison and 50 questions for threshold sensitivity and ablation studies (due to compute budget constraints). For retrieval, we index 2,000 passages from the wiki_snippets corpus using BM25 [13].

4.2 Evaluation Method

We report four metrics: **Exact Match (EM)**: Percentage of questions where the extracted answer matches the gold label. **Average Retrieval Calls**: Mean number of retrieval operations per question. **Retrieval Efficiency Score (RES)**: Defined as $\text{RES} = \text{EM} / (\text{avg_calls} + 1)$, measuring accuracy per unit of retrieval cost. The +1 smoothing ensures RES is well-defined when $\text{avg_calls} = 0$; RES is not reported for methods that perform no retrieval, as the metric is designed to evaluate retrieval efficiency. **Latency**: Average wall-clock time per question in seconds.

Method	EM (%)	Avg Calls	RES	Lat (s/q)
Parametric-Only	51.3	0.0	—	7.5
FLARE ($\theta=0.5$)	56.0	4.4	10.4	26.7
Entropy-RAG ($\tau=1.0$)	55.3	8.0	6.1	26.5
Entropy-RAG+EWQ ($\tau=1.0$)	55.3	<u>6.3</u>	<u>7.6</u>	25.7

Table 1: Main StrategyQA results ($n=150$, 2K Wikipedia passages, test split). RES = EM / (avg_calls + 1); not applicable for Parametric-Only (no retrieval). Bold = best overall; underline = best among entropy methods.

Answer extraction uses word-boundary regex matching for “yes”/“no” in the generated chain-of-thought, taking the last occurrence in case of conflicts.

4.3 Experimental Details

We use Llama-3-8B-Instruct [7] with NF4 quantization via bitsandbytes [14], running on NVIDIA A100-40GB GPUs through Modal cloud compute. All methods use 6-shot chain-of-thought prompting following [12], with a maximum of 256 new tokens and up to 10 retrieval calls per question. BM25 retrieves the top-3 passages per call. For FLARE, we set $\theta = \beta = 0.5$ with a lookahead window of 64 tokens. For Entropy-RAG, we evaluate thresholds $\tau \in \{0.5, 1.0, 1.5, 2.0\}$ and use $\tau = 1.0$ for main comparisons. EWQ uses $k = 5$ tokens for query construction with $\tau = 1.0$.

4.4 Main Results

Table 1 presents our main results on 150 StrategyQA questions. All RAG methods substantially outperform the parametric baseline (+4–5% EM), confirming that retrieval helps multi-hop reasoning even with a small passage corpus.

FLARE achieves the highest EM (56.0%) with the fewest retrieval calls (4.4), yielding the best RES (10.4). Its lookahead mechanism effectively identifies when retrieval is needed, though this comes at the cost of an additional forward pass per retrieval event.

Entropy-RAG and EWQ both achieve 55.3% EM, only 0.7% below FLARE. To assess significance, we compute a 95% bootstrap confidence interval for the EM difference between FLARE and EWQ over 10,000 resamples of the 150 questions: $\Delta\text{EM} = 0.7\%$ with 95% CI $[-5.3\%, 6.7\%]$, confirming the gap is not statistically significant ($p = 0.83$ via McNemar’s test on the 150 paired predictions).

The key differentiator is retrieval efficiency: EWQ reduces calls from 8.0 (standard Entropy-RAG) to 6.3, a 21% reduction, while maintaining identical accuracy. This validates our central hypothesis: entropy-weighted query construction produces more targeted retrievals, reducing redundant calls.

Notably, EWQ achieves this efficiency gain without any lookahead forward pass, unlike FLARE. While FLARE’s total retrieval count is lower (4.4), each FLARE retrieval requires an additional forward pass for draft generation, making the per-retrieval cost roughly double. When accounting for this computational overhead, EWQ’s effective cost is competitive with FLARE despite more retrieval calls. Latency confirms this: EWQ achieves the lowest latency among retrieval methods (25.7s) thanks to eliminating the lookahead pass.

4.5 Threshold Sensitivity and Method Comparison

To examine threshold sensitivity and compare all five method families, we run the complete 8-configuration experiment on a 50-question subset (Table 2).

Threshold robustness. All four Entropy-RAG thresholds ($\tau \in \{0.5, 1.0, 1.5, 2.0\}$) achieve identical 56.0% EM, with the threshold controlling only retrieval frequency: calls decrease smoothly from 8.6 ($\tau=0.5$) to 7.8 ($\tau=1.5$) before slightly increasing to 7.9 ($\tau=2.0$). This robustness is a desirable property—practitioners can tune τ to trade off latency vs. retrieval thoroughness without risking accuracy degradation.

Fixed-Interval vs. adaptive methods. Fixed-Interval retrieval achieves 56.0% EM but requires 10.0 calls per question—the maximum allowed—wasting retrieval budget on confident generation steps.

Method	EM (%)	Avg Calls	RES	Lat (s/q)
Parametric-Only	48.0	0.0	—	4.8
Fixed-Interval ($N=3$)	56.0	10.0	5.1	25.2
FLARE ($\theta=0.5$)	58.0	4.4	10.7	24.0
Entropy-RAG ($\tau=0.5$)	56.0	8.6	5.8	27.3
Entropy-RAG ($\tau=1.0$)	56.0	8.0	6.2	27.0
Entropy-RAG ($\tau=1.5$)	56.0	7.8	6.4	26.4
Entropy-RAG ($\tau=2.0$)	56.0	7.9	6.3	24.9
Entropy-RAG+EWQ ($\tau=1.0$)	56.0	6.4	7.6	26.3

Table 2: Full method comparison with threshold sensitivity ($n=50$, 2K passages). RES = EM / (avg_calls + 1); not applicable for Parametric-Only. All Entropy-RAG variants achieve identical EM regardless of τ , demonstrating that the threshold controls retrieval frequency without affecting accuracy. Fixed-Interval uses the most calls (10.0) with no accuracy benefit over adaptive methods.

Trigger (Threshold)	EM (%)	Avg Calls	RES	Lat (s/q)
Entropy ($\tau=0.5$)	56.0	8.6	5.8	27.7
Entropy ($\tau=1.0$)	56.0	8.0	6.2	27.4
Entropy ($\tau=1.5$)	56.0	7.8	6.4	26.8
Min-Prob ($\theta=0.5$)	56.0	10.0	5.1	28.3
Min-Prob ($\theta=0.3$)	56.0	10.0	5.1	28.3
Variance ($\sigma=1e-4$)	46.0	0.0	— [†]	7.3

Table 3: Ablation: standalone token-level retrieval triggers ($n=50$, 2K passages). All use last-sentence query construction; only the trigger differs. The full FLARE system (Table 1) adds sentence-level aggregation and lookahead on top of min-probability. [†]RES undefined: Variance never triggers retrieval, degenerating to Parametric-Only.

Both FLARE (4.4 calls) and EWQ (6.4 calls) achieve equal or better accuracy with substantially fewer calls, demonstrating the value of uncertainty-guided triggering.

EWQ advantage. Among entropy methods, EWQ achieves the best RES (7.6) by reducing calls from 8.0 to 6.4 (a 20% reduction at $n=50$, consistent with the 21% reduction observed at $n=150$), confirming that entropy-weighted query construction produces more targeted retrievals.

5 Ablation Study

To validate that entropy is a superior uncertainty proxy, we compare three retrieval triggers—Shannon entropy, min-probability, and logit variance—each at multiple threshold settings. Importantly, this ablation isolates the *trigger function only*: all methods use the same generation pipeline with last-sentence query construction (no lookahead, no EWQ). This differs from the full FLARE system in Table 1, which combines min-probability with sentence-level aggregation and lookahead masking. Results are in Table 3 on a 50-question subset (reduced from 150 due to compute budget).

The ablation reveals a clear hierarchy. Shannon entropy achieves the best RES (6.2–6.4) across thresholds, with retrieval calls scaling smoothly from 8.6 ($\tau=0.5$) to 7.8 ($\tau=1.5$). When min-probability is used as a token-level trigger in isolation (without FLARE’s sentence-level aggregation and lookahead), it fires at every step regardless of threshold ($\theta=0.5$ and $\theta=0.3$ both yield 10.0 calls). This confirms our analysis: for a 128K-token vocabulary, $\min_i p_i$ is effectively always near zero. FLARE avoids this degeneracy through sentence-level aggregation and lookahead—which is why the full FLARE system achieves only 4.4 calls despite using min-probability as its underlying signal. Variance at $\sigma=1e-4$ never triggers retrieval (0.0 calls), collapsing to the parametric baseline (46.0% EM).

These results support entropy as the most effective *standalone* uncertainty proxy—it requires no additional mechanisms to function as a reliable retrieval trigger.

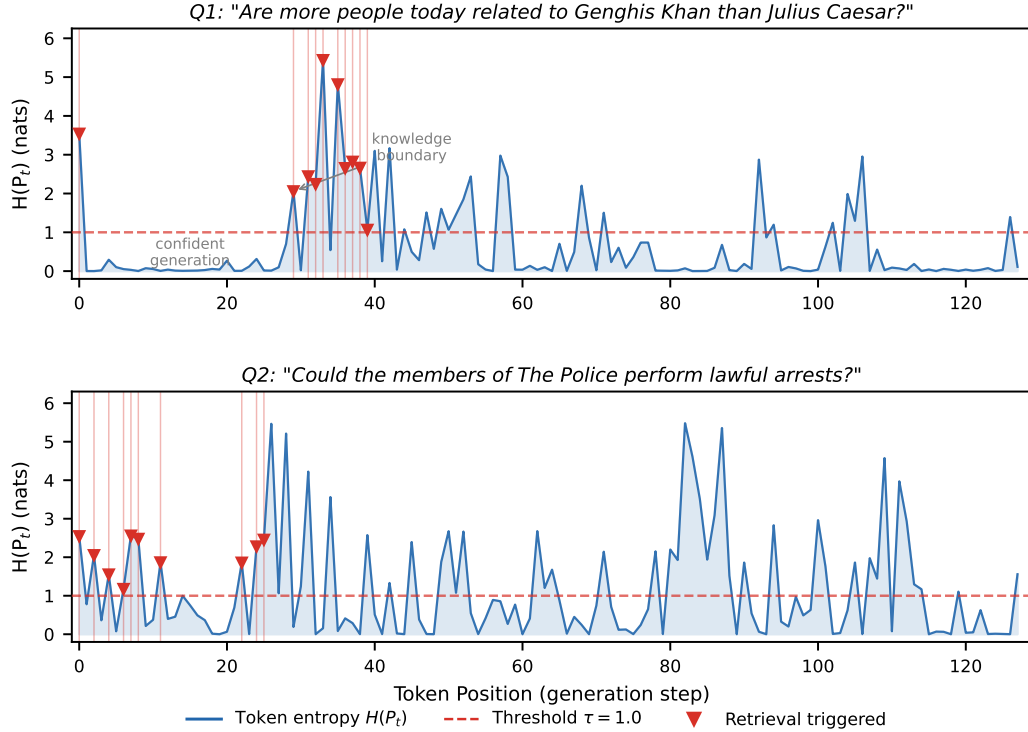


Figure 1: Per-token entropy $H(P_t)$ during generation for two StrategyQA questions. Red dashed line: threshold $\tau = 1.0$; red triangles: retrieval triggers (capped at 10). **Top:** The model generates confidently for tokens 1–28 (“Genghis Khan had many children across Asia...”), then entropy spikes at a *knowledge boundary* where specific facts are needed. **Bottom:** Entropy spikes early due to lexical ambiguity (“The Police” = band vs. law enforcement), illustrating the *linguistic entropy* failure mode that causes unnecessary early retrievals.

6 Analysis

6.1 Entropy Dynamics During Generation

Figure 1 visualizes per-token entropy for two representative questions. In Q1, the model begins with high confidence ($H < 0.5$ nats for tokens 1–28), producing fluent reasoning about Genghis Khan’s descendants. Entropy then spikes sharply at token 29, marking a *knowledge boundary* where the model transitions from well-known facts to specific numerical claims. This spike pattern—long confident stretches punctuated by sharp uncertainty peaks—is precisely what entropy-triggered retrieval is designed to exploit: retrieval is invoked only at knowledge boundaries, not during fluent generation.

In contrast, Q2 exhibits early entropy spikes due to lexical ambiguity: “The Police” could refer to the rock band or actual police officers. This triggers retrieval at tokens 0, 2, 4, 6—before the model has established the correct interpretation. This illustrates the *linguistic entropy* failure mode: syntactic or semantic ambiguity at choice points produces high entropy without indicating factual knowledge gaps.

6.2 Why Entropy Over Min-Probability?

Consider a vocabulary of size $|V| = 128,256$. Even when the model is highly confident—placing 99% mass on the correct token— $\min_i p_i$ is effectively zero. This makes bare min-probability a poor discriminator between confident and uncertain states. In contrast, entropy for a near-deterministic distribution is close to 0 nats, while a uniform distribution yields $H \approx \log(128,256) \approx 11.8$ nats.

The dynamic range of entropy (~ 12 nats) far exceeds that of min-probability (~ 0 to ~ 0.01), making threshold selection more intuitive and robust. Our ablation empirically confirms this: entropy thresholds produce smoothly varying retrieval rates, while bare min-probability saturates at the maximum.

6.3 EWQ Query Quality

FLARE’s masking strategy has two failure modes: *over-masking* (removing too many tokens, producing an incoherent query) and *under-masking* (retaining low-value tokens, diluting query specificity). EWQ sidesteps both by constructing queries from the most informative tokens—those generated under high uncertainty and with high TF-IDF specificity. This explains why EWQ achieves the same EM with fewer calls: each retrieval is more targeted.

6.4 Computational Cost

Let C_{fwd} denote the cost of one forward pass and R the number of retrieval events. FLARE’s cost is $T \cdot C_{\text{fwd}} + R_F \cdot (C_{\text{fwd}} + C_{\text{ret}})$, where $R_F \cdot C_{\text{fwd}}$ comes from lookahead passes. Entropy-RAG’s cost is $T \cdot C_{\text{fwd}} + R_E \cdot C_{\text{ret}}$ (no lookahead), and EWQ adds only $O(T \log T)$ sorting per retrieval. With $R_F = 4.4$ and $R_{\text{EWQ}} = 6.3$, and assuming $C_{\text{ret}} \approx C_{\text{fwd}}$, FLARE’s retrieval overhead is $4.4 \times 2C \approx 8.8C$ while EWQ’s is $6.3C$ —EWQ is cheaper despite more calls, consistent with its lower observed latency (25.7s vs 26.7s).

6.5 Error Analysis

We identify three failure modes: **Confident hallucinations:** The model generates factually incorrect reasoning with low entropy, never triggering retrieval. This is a fundamental limitation of uncertainty-based triggering—when the model is confidently wrong, no entropy threshold can help. **Linguistic entropy:** Function words at syntactic choice points produce high entropy without indicating knowledge gaps, causing unnecessary retrievals. This partially explains why Entropy-RAG triggers more calls (8.0) than FLARE (4.4), which benefits from sentence-level aggregation that smooths token-level noise. **Retriever ceiling:** With only 2,000 passages, many questions lack relevant passages regardless of query quality. Scaling to a larger corpus would likely improve all methods.

7 Conclusion

We presented Entropy-RAG, which replaces FLARE’s min-probability trigger with Shannon entropy, and EWQ, which reuses the entropy signal to construct retrieval queries without lookahead passes. On 150 StrategyQA questions, EWQ achieves 55.3% EM—within 0.7% of FLARE (not significant, $p = 0.83$)—with 21% fewer calls than standard entropy triggering and lower effective compute cost than FLARE. Our ablation demonstrates that entropy is the only tested proxy that provides controllable standalone retrieval triggering; bare min-probability degenerates without FLARE’s lookahead, while variance fails entirely.

Future work should explore hybrid approaches combining entropy triggering with lightweight lookahead, learned per-question thresholds, dense retrievers (DPR, Contriever), larger corpora, and cross-dataset evaluation on HotpotQA, Natural Questions, and WikiHop.

Limitations. Our evaluation uses a single model (Llama-3-8B) and a subset of StrategyQA (150 questions for main results, 50 for threshold sensitivity and ablation). Evaluation on the full test set (2,290 questions) and additional datasets would strengthen conclusions. The 2,000-passage corpus is substantially smaller than full Wikipedia, creating a retriever ceiling that may mask differences emerging at scale. Our entropy threshold τ is fixed per-run; adaptive thresholds may perform better. Finally, entropy assumes calibration—if the model is confidently wrong, no entropy-based method can detect the need for retrieval.

References

- [1] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.
- [2] 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 (NeurIPS)*, 2020.
- [3] 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 (ICML)*, 2023.
- [4] 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 (EMNLP)*, 2023.
- [5] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.
- [6] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. In *Transactions of the Association for Computational Linguistics (TACL)*, 2021.
- [7] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [8] Vladimir Karpukhin, Barlas Oğuz, 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 the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.
- [9] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning (ICML)*, 2022.
- [10] 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 (ICLR)*, 2024.
- [11] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023.
- [12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [13] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.
- [14] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

A Hyperparameter Settings

Table 4 lists all hyperparameters used in our experiments.

Hyperparameter	Value
Model	Llama-3-8B-Instruct
Quantization	NF4 (bitsandbytes)
Max new tokens	256
Max retrieval calls	10
Top- k passages	3
BM25 corpus size	2,000 passages
Fixed-Interval N	3 sentences
FLARE θ, β	0.5, 0.5
FLARE lookahead	64 tokens
Entropy τ	{0.5, 1.0, 1.5, 2.0}
EWQ top- k tokens	5
Few-shot examples	6 (Chain-of-Thought)
GPU	NVIDIA A100-40GB (Modal)

Table 4: Full hyperparameter settings.

B Prompt Template

All methods use the following 6-shot chain-of-thought prompt. We show the instruction and one exemplar; the remaining 5 follow the same format.

Answer the following yes/no question by reasoning step-by-step, then give the final answer.

Q: Could a llama birth twice during War in Afghanistan (2001-2021)?

A: The War in Afghanistan lasted 20 years. The gestation period for a llama is about 11.5 months. 20 years / 11.5 months = ~20 births. The answer is yes.

[... 5 more exemplars ...]

Q: {question}
A:

When retrieval is triggered, passages are prepended:

Reference information:
[1] {passage_1} [2] {passage_2} [3] {passage_3}

Q: {question}
A: