Modeling Uncertainty Trends for Timely Retrieval in Dynamic RAG

Bo Li[1, 3], **Tian Tian**[1], **Zhenghua Xu**[1*], **Hao Cheng**[2], **Shikun Zhang**[3], **Wei Ye**[3*]

[1]State Key Laboratory of Intelligent Power Distribution Equipment and System,
School of Health Sciences and Biomedical Engineering, Hebei University of Technology, China
[2]School of Artificial Intelligence, Hebei University of Technology, China
[3]National Engineering Research Center for Software Engineering, Peking University, China

Abstract

Dynamic retrieval-augmented generation (RAG) allows large language models (LLMs) to fetch external knowledge on demand, offering greater adaptability than static RAG. A central challenge in this setting lies in determining the optimal timing for retrieval. Existing methods often trigger retrieval based on low token-level confidence, which may lead to delayed intervention after errors have already propagated. We introduce Entropy-Trend Constraint (ETC), a training-free method that determines optimal retrieval timing by modeling the dynamics of token-level uncertainty. Specifically, ETC utilizes first- and second-order differences of the entropy sequence to detect emerging uncertainty trends, enabling earlier and more precise retrieval. Experiments on six QA benchmarks with three LLM backbones demonstrate that ETC consistently outperforms strong baselines while reducing retrieval frequency. ETC is particularly effective in domain-specific scenarios, exhibiting robust generalization capabilities. Ablation studies and qualitative analyses further confirm that trend-aware uncertainty modeling yields more effective retrieval timing. The method is plug-and-play, model-agnostic, and readily integrable into existing decoding pipelines. Implementation code is included in the supplementary materials.

Code — https://github.com/pkuserc/ETC

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as a powerful paradigm for augmenting large language models (LLMs) with external knowledge, effectively addressing limitations such as outdated training data and narrow domain coverage (Gao et al. 2023; Kandpal et al. 2023; Mousavi, Alghisi, and Riccardi 2024; Xiong et al. 2024). By incorporating retrieved documents during generation, RAG systems significantly improve factual accuracy and enhance generalization across domains (Xu et al. 2024; Fang et al. 2024). While early RAG systems typically performed a single retrieval at the start of generation (Wang et al. 2024a; Shi et al. 2023; Wang, Yang, and Wei 2023; Yu et al. 2023), recent developments have introduced dynamic RAG, where retrieval is triggered conditionally during decoding to balance informativeness and efficiency (Jiang et al. 2023; Su et al. 2024b).

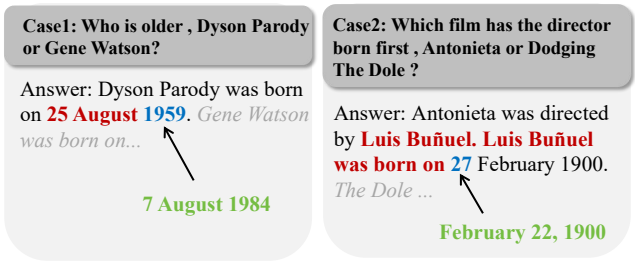

Figure 1: The delayed retrieval issue exists in current dynamic RAG method, where blue tokens represent DRAGIN's retrieval timing, and red tokens highlight incorrectly generated tokens caused by delayed retrieval.

A central challenge in dynamic RAG is deciding when retrieval should occur. Previous approaches mainly rely on token-level uncertainty heuristics. For instance, some methods (Borgeaud et al. 2022; Trivedi et al. 2022; Ram et al. 2023) perform retrieval after a fixed number of tokens or sentences, while others trigger retrieval when the confidence of a newly generated token drops below a predefined threshold (Jiang et al. 2023; Su et al. 2024b). Although intuitive, such reactive mechanisms often suffer from delayed retrieval, i.e., retrieving only after the model has already deviated from the correct generation path. As illustrated in Figure 1, retrieval triggered too late may fail to prevent factual errors, whereas overly early or frequent retrieval increases latency and redundancy (Ni et al. 2024; Ren et al. 2023; Chen et al. 2024; Maekawa et al. 2024).

We argue that retrieval timing should be guided not by isolated token-level confidence values, but by tracking the overall trend of uncertainty throughout the generation process. Foundational studies on LLMs show that entropy-based uncertainty measures are more robust and informative than pointwise confidence for detecting hallucinations or unreliable generation. For instance, recent work has employed entropy as a signal for token-level factuality assessment, semantic instability, and fine-grained uncertainty estimation (Fadeeva et al. 2024; Farquhar et al. 2024; Nikitin et al. 2024). These findings suggest that tracking how uncertainty evolves over time yields more reliable signals than reacting to isolated confidence drops at individual tokens.

*Corresponding Authors.

This observation points to a promising direction: modeling the dynamics of token-level uncertainty to improve retrieval decisions during generation.

Building on this insight, we propose **E**ntropy-**T**rend **C**onstraint (**ETC**), a novel training-free method that models the dynamics of uncertainty throughout generation rather than relying on individual token-level confidence. Specifically, ETC analyzes the first- and second-order differences of the token-level entropy sequence to detect emerging low-confidence patterns before they become critical. These differential operations are classical tools for discrete sequence analysis (Jordán 1965; Levy and Lessman 1992; Ames 2014). In particular, the second-order difference provides a sensitive signal for detecting rapid shifts in entropy, indicating that the model may be entering an unstable prediction phase. To enhance robustness, we further introduce a dynamic smoothing mechanism to reduce the impact of entropy outliers and stabilize retrieval decisions. By leveraging confidence trends, ETC enables timely retrieval, injecting external knowledge at more optimal positions while reducing retrieval frequency. Unlike existing methods that rely on heuristic rules or costly training procedures, ETC is plug-and-play, model-agnostic, and easily integrable into any autoregressive decoding pipeline.

We evaluate ETC on six diverse benchmarks spanning multi-hop reasoning, commonsense QA, reading comprehension, and biomedical QA. Across three LLM backbones, ETC consistently outperforms strong baselines while requiring fewer retrieval operations and achieving significantly lower rates of delayed and redundant retrieval. Furthermore, qualitative evaluations using GPT-4o and extensive ablation studies validate the precision and efficiency of ETC's retrieval timing strategy.

- We identify and systematically analyze the delayed retrieval problem in dynamic RAG, revealing fundamental limitations in confidence-based triggering strategies.
- We propose **E**ntropy-**T**rend **C**onstraint (ETC), a novel training-free retrieval strategy that leverages uncertainty dynamics for timely and efficient knowledge injection.
- Experiments on six diverse benchmarks with three LLM backbones demonstrate that ETC consistently improves performance while reducing retrieval frequency. We further present comprehensive analyses and case studies to support these findings.

2 Preliminary and Delayed Retrieval

In this section, we briefly introduce the fundamentals of RAG and the issue of delayed retrieval.

2.1 Preliminary

In a standard RAG system, for a given query q, a retriever r retrieves a set of relevant documents $C = \{c_1, c_2, .., c_n\}$ from a large corpus D. During inference, the query q and retrieved contexts C are combined through a prompt p to form a new input for a given LLM, which then generates the retrieval-augmented output y. Typically, retrieval is performed once at the beginning of the generation process (Melz 2023; Wang et al. 2024a; Li, Yuan, and Zhang 2024). The mathematical expression of this process is shown as follows:

$$y = LLM(q, C, p). \tag{1}$$

However, studies have shown that retrieving information solely at the beginning of generation may not always be optimal, as irrelevant or redundant context can introduce noise and hinder model performance. An alternative approach is Dynamic RAG, which performs retrieval only when needed. Recent methods typically trigger retrieval when the model generates a token with very low confidence. Let time t denote the point at which retrieve is triggered, indicating that external knowledge is required after generating the t-th token. The prompt at the step t is denoted as p_t. The dynamic RAG model can then be formally expressed as:

$$\hat{y} = LLM(q, C_t, p_t, y_{<t}), \tag{2}$$

where $y_{<t}$ represents the sequence of tokens generated before time t, and C_t is the external knowledge retrieved after the t-th token has been generated.

2.2 Delayed Retrieval

While using the confidence of a single token to trigger retrieval is a straightforward approach, it does not always lead to optimal retrieval timing. Through both qualitative and quantitative analyses, we identify a delayed retrieval problem that arises when retrieval timing is determined primarily by token-level confidence. Specifically, we examine cases from the 2WikiMultihopQA (Ho et al. 2020) dataset using the DRAGIN framework (Su et al. 2024b). As shown in Figure 1, retrieval is triggered when a generated token falls below DRAGIN's predefined confidence threshold. These examples clearly illustrate that by the time retrieval is triggered, the generation has already deviated from the correct path. Consequently, this approach often results in multiple low-confidence tokens being generated before retrieval, suggesting that confidence-based intervention may not be the most effective solution.

In addition, we manually annotate 100 instances to compute the proportion of delayed retrieval, and find that around 33% of them exhibit this issue (see **Section** 6.3 for more details). These findings highlight the need for approaches that are more sensitive to token-level confidence trends during generation.

3 Proposed Method

Building on the above analysis, we propose **E**ntropy-**T**rend **C**onstraint (**ETC**), a novel dynamic RAG method that mitigates delayed retrieval by considering the confidence trend of the generated sequence. Specifically, ETC computes confidence trends based on the entropy sequence and triggers retrieval when confidence changes sharply. In addition, we introduce a dynamic smoothing strategy that reduces unnecessary retrievals while maintaining model performance.

3.1 Entropy-Trend Constraint

Given an input c and a prompt p, a LLM generates an output token sequence, denoted as $T = \{t_1, t_2, ..., t_n\}$, where the

length of T is n. For each generated token t_i, we compute its prediction distribution $p_i(v)$ over the vocabulary $\mathcal{V}$, and use entropy as a measure of its prediction uncertainty, defined as follows:

$$\mathcal{H}_i = -\sum_{v \in \mathcal{V}} p_i(v) \log p_i(v). \tag{3}$$

Entropy is widely used in natural language processing tasks to quantify the uncertainty of a given probability distribution. A lower entropy value indicates higher confidence in the LLM's prediction. For the generated output T, we define its entropy sequence $\mathcal{H}$ as follows:

$$\mathcal{H} = \{\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_n\}. \tag{4}$$

The entropy sequence $\mathcal{H}$ reflects the confidence associated with each generated token. However, static entropy values alone do not capture dynamic fluctuations during the generation process. Therefore, we utilize both first and second differences of the entropy sequence. The first difference is defined as the difference between consecutive terms. This operation measures the change between successive elements in the entropy sequence and is useful for identifying trends and linearity. For the entropy sequence $\mathcal{H}$ of the generated tokens, the first difference is computed as bellow:

$$\Delta\mathcal{H} = \{\Delta\mathcal{H}_1, \Delta\mathcal{H}_2, ..., \Delta\mathcal{H}_{n-1}\}, \tag{5}$$

$$\Delta\mathcal{H}_i = \mathcal{H}_{i+1} - \mathcal{H}_i. \tag{6}$$

For example, $\Delta\mathcal{H}_1 = \mathcal{H}_2 - \mathcal{H}_1$, and $\Delta\mathcal{H}$ has $n-1$ items. While the first difference reflects the variation between adjacent time points, it does not reveal how rapidly these changes occur. Monitoring the rate of change in these trends is crucial for timely retrieval, as it helps detect early signs of instability in the model's confidence. Thus we further compute the second difference of the entropy sequence $\mathcal{H}$ to capture the rate of confidence change:

$$\Delta^2\mathcal{H} = \Delta(\Delta\mathcal{H}) = \{\Delta^2\mathcal{H}_1, \Delta^2\mathcal{H}_2, ..., \Delta^2\mathcal{H}_{n-2}\}, \tag{7}$$

$$\Delta^2\mathcal{H}_i = \Delta\mathcal{H}_{i+1} - \Delta\mathcal{H}_i. \tag{8}$$

Alternatively, we can also obtain the $\Delta^2\mathcal{H}$ based on $\mathcal{H}$ directly, where:

$$\Delta^2\mathcal{H}_i = \mathcal{H}_{i+2} - 2\mathcal{H}_{i+1} + \mathcal{H}_i. \tag{9}$$

The second difference highlights rapid shifts in confidence and thus serves as a more sensitive indicator for triggering retrieval.

3.2 Dynamic Smoothing Method

While the second difference improves retrieval timing, it can still be influenced by outlier entropy values from specific tokens, potentially leading to redundant retrievals. To address this issue, we propose a dynamic smoothing method that reduces the impact of outliers by assigning them lower weights during aggregation. Specifically, dynamic smoothing computes the $\Delta^2\widehat{\mathcal{H}}_t$ using weighted average of $\Delta^2\mathcal{H}_t$ and $\Delta^2\mathcal{H}_{t-1}$, the weight w_t of $\Delta^2\mathcal{H}_t$ is obtained by the following equation:

$$w_t = \frac{|\Delta^2\mathcal{H}_{t-1} - E_t|}{|\Delta^2\mathcal{H}_t - E_t| + |\Delta^2\mathcal{H}_{t-1} - E_t|}, \tag{10}$$

$$E_t = \mathbb{E}[\Delta^2\mathcal{H}_1, \Delta^2\mathcal{H}_2, ..., \Delta^2\mathcal{H}_t]. \tag{11}$$

Here, E_t denotes the mathematical expectation of the entropy sequence's second difference $\{\Delta^2\mathcal{H}_1, \Delta^2\mathcal{H}_2, ..., \Delta^2\mathcal{H}_t\}$. A higher $\Delta^2\mathcal{H}_t$ results in a relatively lower w_t, thereby reducing the impact of outlier entropy and producing a smoothed $\Delta^2\widehat{\mathcal{H}}_t$ at the current timing t. The smoothed $\Delta^2\widehat{\mathcal{H}}_t$ is computed as follows:

$$\Delta^2\widehat{\mathcal{H}}_t = w_t\Delta^2\mathcal{H}_t + w_{t-1}\Delta^2\mathcal{H}_{t-1}, \tag{12}$$

By mitigating the influence of the outlier entropy, the smoothed second difference $\Delta^2\widehat{\mathcal{H}}_t$ reduces unnecessary retrievals. We perform a retrieval operation at time t if $|\Delta^2\widehat{\mathcal{H}}_t| \geq \alpha$, where α is a predefined threshold and will be tuned on the validation set.

3.3 Query Construction and Continue Generation

Once ETC determines the optimal retrieval timing, the next challenge is to construct an effective query based on the original input and the generated text so far. We adopt the query construction method from DRAGIN, which leverages the self-attention mechanism of Transformer-based LLMs. This approach ranks tokens by their attention scores and selects the *top-n* tokens to form the query[1]. After constructing the query at timestep t, ETC retrieves relevant information $C_t = \{C_t^1, C_t^2, \dots\}$ from the external corpus, where C_t^i denotes the i-th retrieved document.

To continue generation after retrieval, we combine the original query q, the previously generated text $y_{<t}$, the retrieved information C_t, and the prompt p_t to guide the LLM in generating the subsequent output. The generated token y_t serves as the prefix for subsequent generation, ensuring that the output remains coherent and consistent with the prior sequence $y_{<t}$.

4 Experimental Setup

4.1 Dataset and Evaluation Metic

To comprehensively evaluate the effectiveness of ETC across diverse scenarios, we conduct experiments on six representative datasets: 2WikiMultihopQA (Ho et al. 2020), HotpotQA (Yang et al. 2018), StrategyQA (Geva et al. 2021), IIRC (Ferguson et al. 2020), BioASQ (Tsatsaronis et al. 2015), and PubMedQA (Jin et al. 2019). The domains and evaluation metrics for each dataset are summarized in Table 1. Specifically, the first four datasets are used for general-purpose multi-hop and commonsense QA evaluation, while the last two assess ETC's effectiveness in settings that require biomedical domain knowledge under limited-resource conditions.

[1]Due to space limitations, please refer to Su et al. (2024b) for further details.

Dataset	Domain	Metric
2WikiMultihopQA	Multi-hop	EM, F1
HotpotQA	Multi-hop	EM, F1
StrategyQA	Commonsense	Accuracy
IIRC	Reading	EM, F1
BioASQ	Biomedical	Accuracy
PubMedQA	Biomedical	Accuracy

Table 1: The statistics of the datasets used in this study, and we adopt the same metrics as previous works.

4.2 Experimental Details

To ensure a fair comparison with previous works, all experimental settings are the same as FLARE (Jiang et al. 2023) and DRAGIN (Su et al. 2024b). Specifically, we follow the setting of Wang et al. (2022) to generate both chain-of-thought (CoT) reasoning process as well as the final answer, we also use prompt templates from Trivedi et al. (2022); Jiang et al. (2023); Wei et al. (2022) tailored to each dataset. We use BM25 as the retriever due to its high efficiency and strong retrieval performance. To compute the entropy sequence, we remove stop words using the SpaCy library [2]. Wikipedia is used as the external knowledge corpus, from which we retrieve three augmented passages at each retrieval step. The backbone LLMs include LLaMa2-7b, LLaMa2-13b (Touvron et al. 2023), LLaMa3-8b (Grattafiori et al. 2024) and Vicuna-13b-v1.5 (Chiang et al. 2023), we report the results of LLaMa2-13B in the Appendix due to space limitations.

4.3 Comparison Models

Since ETC is a training-free dynamic RAG method, it does not need any pre-training or fine-tuning process. In this paper, we compare ETC with the following advanced training-free RAG methods. 1) **w/o RAG** directly asks LLMs to generate answers without any retrieval operation; 2) **Single RAG** retrieves augmented information only once at the beginning of the generation based on the initial question; 3) **In-Context RALM** (Ram et al. 2023) triggers the retrieval module every n tokens; 4) **IRCoT** (Trivedi et al. 2022) activates the retrieval module every sentence; 5) **FLARE** (Jiang et al. 2023) conducts retrieve when a token's uncertainty below a threshold; 6) **DRAGIN** (Su et al. 2024b) further considers both the importance and uncertainty of the generated token to determine the retrieval timing.

Besides the above previous works, we built several model variants for ablation study: 1) **ETC**$_{1st}$ indicates that we use the first difference and dynamic smoothing method to determine retrieval timing; 2) **ETC w/o smoothing** removes the dynamic smoothing method and relies solely on the second difference of the entropy sequence to trigger retrieval; 3) **ETC**$_{fixed}$ uses a fixed weight factor w_t to smooth the second difference.

[2]https://spacy.io/

5 Main Results

5.1 Main Results

Table 2 presents the main results, from which we draw three key findings. (1) Simple training-free RAG methods, such as In-Context RALM and IRCoT, which perform retrieval at fixed intervals (e.g., every n tokens or each sentence), fail to consistently improve performance across tasks or models. This supports prior findings that indiscriminate or misaligned retrieval may degrade generation quality. (2) ETC consistently achieves the best overall performance across all evaluated settings. It outperforms both static and dynamic RAG baselines, achieving the highest average scores on each model: 0.344 on LLaMA2-7B, 0.420 on LLaMA3-8B, and 0.376 on Vicuna-13B, with relative improvements ranging from 5.9% to 12.1% over the strongest competing methods. These results validate that ETC's trend-based retrieval mechanism enables more accurate and timely intervention compared to token-level confidence thresholds. (3) ETC further demonstrates strong adaptability across different task types and model scales. Notably, on LLaMA3-8B, the Single RAG baseline achieves competitive performance (0.375), surpassing several dynamic RAG methods such as DRAGIN and FLARE. This suggests that in high-capacity models, suboptimal retrieval timing can disrupt the generation process, leading to performance degradation. In contrast, ETC maintains robust improvements (0.420), indicating that its trend-aware strategy scales effectively with stronger LLMs.

Overall, ETC mitigates delayed retrieval by modeling confidence trends, thereby enabling more effective integration of external knowledge and improving generation performance.

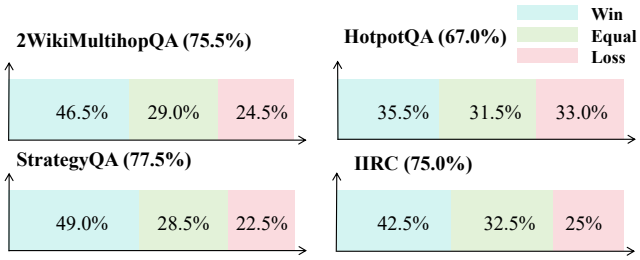

Figure 2: The win rate using GPT-4o as judge. The value in each bracket indicates the percentage of times ETC's answer quality is equal to or better than DRAGIN's on the corresponding dataset.

5.2 Win Rate

Several studies (Li et al. 2023; Yang et al. 2024; Li et al. 2024) suggest that traditional metrics such as EM and F1 alone may not fully capture the performance of generative models, as these models can produce semantically correct answers that do not exactly match human-labeled references at the token level. To better assess answer quality, we employ GPT-4o as an evaluator. Specifically, we randomly sample 200 instances from each dataset and ask GPT-4o to judge which answer is more reasonable based on the given question and the ground-truth answer. The evaluation prompt in-

LLM	RAG Method	2WikiMultihopQA		HotpotQA		StrategyQA	IIRC		
		EM	F1	EM	F1	Accuracy	EM	F1	Avg.Score
Llama2-7b	**w/o RAG**	0.146	0.223	0.184	0.275	**0.659**	0.139	0.173	0.257
	Single RAG	0.169	0.255	0.164	0.250	0.645	0.187	0.226	0.271
	In-Context RALM	0.112	0.192	0.146	0.211	0.635	0.172	0.202	0.239
	IRCoT	0.189	0.265	0.214	0.304	0.630	0.178	0.216	0.285
	FLARE	0.143	0.213	0.149	0.221	0.627	0.136	0.164	0.236
	DRAGIN	0.220	0.293	0.232	0.334	0.641	0.192	0.234	0.307
	ETC(Ours)	**0.271**	**0.360**	**0.288**	**0.401**	0.650	**0.199**	**0.240**	**0.344**$_{+12.1\%}$
Llama3-8b	**w/o RAG**	0.174	0.258	0.281	0.379	0.667	0.187	0.223	0.310
	Single RAG	0.241	0.349	**0.339**	0.454	0.651	0.275	0.316	0.375
	In-Context RALM	0.267	0.376	0.243	0.341	0.641	0.176	0.213	0.322
	IRCoT	0.268	0.376	0.216	0.314	0.613	0.188	0.226	0.314
	FLARE	0.197	0.276	0.246	0.341	0.609	0.183	0.214	0.295
	DRAGIN	0.212	0.302	0.272	0.378	0.662	0.189	0.226	0.320
	ETC(Ours)	**0.352**	**0.453**	0.272	**0.487**	**0.672**	**0.286**	**0.328**	**0.420**$_{+12.0\%}$
Vicuna-13b	**w/o RAG**	0.146	0.223	0.228	0.326	0.682	0.175	0.215	0.285
	Single RAG	0.170	0.256	0.254	0.353	0.686	0.217	0.256	0.313
	In-Context RALM	0.135	0.213	0.187	0.304	0.645	0.099	0.129	0.245
	IRCoT	0.188	0.263	0.185	0.322	0.622	0.103	0.134	0.260
	FLARE	0.157	0.226	0.092	0.181	0.599	0.117	0.147	0.217
	DRAGIN	0.252	0.352	0.288	0.416	0.687	**0.223**	0.265	0.355
	ETC(Ours)	**0.282**	**0.373**	**0.347**	**0.456**	**0.693**	0.216	**0.268**	**0.376**$_{+5.9\%}$

Table 2: The main results of various RAG methods. All results are obtained from publicly available papers or reproduced using open-source code. The best result of each dataset is in bold. To minimize randomness, we tested each model three times and reported the average performance.Additionally, we conducted t-tests to compare our results with previous results, confirming that our results are statistically significant with a p-value of less than 0.05.

structs GPT-4o to assess responses based on accuracy, completeness, fluency, and relevance [3]. The results in Figure 2 show that ETC consistently achieves performance comparable to or better than DRAGIN across most datasets. This confirms that ETC generates more reasonable answers than previous SoTA models, further demonstrating its effectiveness as a dynamic RAG system.

5.3 Domain-Specific Dataset Evaluation

RAG methods are particularly valuable for injecting external knowledge when LLMs lack domain-specific expertise, making them essential for solving domain-specific tasks. To assess their effectiveness in domain-specific scenarios, we evaluate RAG methods on two biomedical datasets: BioASQ and PubMedQA. As shown in Table 3, ETC achieves substantial improvements over existing RAG models with fewer retrievals, demonstrating its effectiveness in domain-specific tasks. These findings suggest that ETC triggers retrieval at the appropriate moment, ensuring external knowledge is injected when needed, thereby reducing hallucinations and enhancing domain-specific understanding.

Method	BioASQ		PubmedQA	
	Acc.	Count.	Acc.	Count.
Single RAG	0.478	1.0	0.485	1.0
IRCoT	0.319	6.038	0.397	6.433
DRAGIN	0.527	1.849	0.510	1.898
ETC	**0.689**$_{+30.7\%}$	1.495	**0.545**$_{+6.9\%}$	1.747

Table 3: Accuracy (**Acc.**) and retrieval count (**Count.**) on domain-specific datasets. We use Vicuna-13b here since it achieves the best results in the main evaluation.

6 Analysis

6.1 Ablation Study

We conduct ablation studies to assess the effectiveness of using the second difference and the dynamic smoothing strategy. Table 4 shows the results and we have the following observations: 1) Compared to DRAGIN, which triggers retrieval based on single-token confidence, and ETC$_{1st}$ which uses only the first difference of the entropy sequence, using the second difference yields significantly better average scores; 2) Removing the dynamic smoothing module or replacing it with a fixed weight factor (e.g., 0.9 in our experiment) results in consistent performance degradation across most datasets. The above results verify the effectiveness of our proposed components in handling various question answering scenarios.

6.2 Retrieval Efficiency

The average number of retrievals serves as a key metric for assessing the efficiency of dynamic RAG methods. As

[3]The full prompt is provided in Appendix A

	2WikiMultihopQA		HotpotQA		StrategyQA	IIRC		Avg.Score
	EM	F1	EM	F1	ACC	EM	F1	
ETC$_{1st}$	0.271	0.364	0.275	0.388	0.658	0.198	0.234	0.341
ETC$_{fixed}$	0.271	0.360	0.261	0.371	0.650	0.194	0.229	0.334
ETC	0.271	0.360	0.288	0.401	0.650	0.199	0.240	**0.344**
ETC w/o smoothing	0.269	0.358	0.269	0.376	0.641	0.193	0.230	0.334

Table 4: Ablation studies on various components in ETC with LLaMA2-7B-chat as backbone, other LLMs show similar results. We set the fixed weight in **ETC**$_{fixed}$ as 0.9 here.

shown in Table 5, ETC consistently requires fewer retrievals than prior dynamic RAG baselines such as DRAGIN and FLARE. Moreover, removing the dynamic smoothing module or replacing it with a fixed weight leads to an increased number of retrievals. This validates that relying solely on entropy trends may result in redundant retrievals. The dynamic smoothing module effectively reduces unnecessary retrievals by mitigating the impact of outliers in the entropy sequence, thereby improving overall RAG efficiency[4]. In addition, we compute average delayed length, and results show that DRAGIN retrieves on average 8.64 tokens later than ETC. This further confirms that ETC is not only more accurate in deciding whether to retrieve, but also significantly more timely.

	WQA	HQA	SQA	IIRC	Avg.R
FLARE	1.21	1.84	1.01	2.37	1.59
DRAGIN	2.67	3.23	4.39	2.96	3.31
ETC$_{fixed}$	1.56	1.07	1.64	1.64	1.47
ETC	1.43	0.88	1.37	1.48	**1.29**
w/o smoothing	1.47	0.92	1.47	1.52	1.34

Table 5: The average retrieval count for each dataset, where **Avg.R** denotes the average retrieval count across datasets and LLMs. To conserve space, we abbreviate 2WikiMultihopQA, HotpotQA, and StrategyQA as WQA, HQA, and SQA, respectively.

6.3 Selecting Optimal Retrieval Timing

We further investigate whether ETC can mitigate the delayed retrieval and the redundant retrievals. To evaluate retrieval timeliness, we randomly select 100 samples from 2WikiMultihopQA and ask three experts to manually assess whether retrieval operations occurred at the appropriate moment. For the **delayed retrieval**, a retrieval operation is considered delayed if incorrect tokens are generated before retrieval occurs; otherwise, it is classified as timely. The results in Table 6 indicate that DRAGIN exhibits high delayed retrieval ratios, while ETC achieves the lowest delay ratio. This is because the second difference effectively captures the rate of confidence change, making it more sensitive to rapid fluctuations. This enables earlier retrieval interventions before low-confidence tokens are generated. As for **redundant retrieval**, a retrieval is considered redundant if the LLM could generate the correct answer even without retrieving external information. We can observe from Table 6 that ETC shows a significantly lower redundant retrieval ratio than DRAGIN and ETC w/o smoothing; this verifies the usefulness of the dynamic smoothing module in improving retrieval efficiency.

[4]We report detailed retrieval counts for each dataset with each LLM in Appendix C.

	Delayed Retrieval Ratio	Redundant Retrieval Ratio
DRAGIN	0.33	0.95
ETC	**0.22**	**0.79**
w/o smoothing	0.22	0.91

Table 6: The manually annotated delayed retrieval ratio and redundant retrieval ratio, and lower is better.

6.4 The Heat-map of Retrieval Timing and The Entropy Distribution

In this subsection, we present a heat-map illustrating the word positions where ETC and DRAGIN trigger their first retrieval. Additionally, we provide the average entropy values for each word position. The visualization shows that ETC typically triggers retrieval earlier than DRAGIN, which aligns with the trend of gradually increasing word entropy. In contrast, DRAGIN often delays retrieval until encountering highly uncertain words or even later.

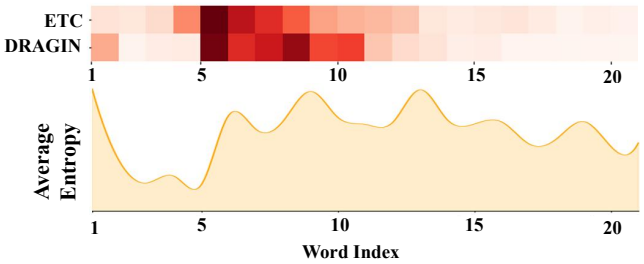

Figure 3: The heat-map of retrieval timing and the entropy distribution.

Interestingly, while DRAGIN appears redder at the first position in the heat-map, which may suggest early retrieval, a closer analysis reveals that most of these early retrievals

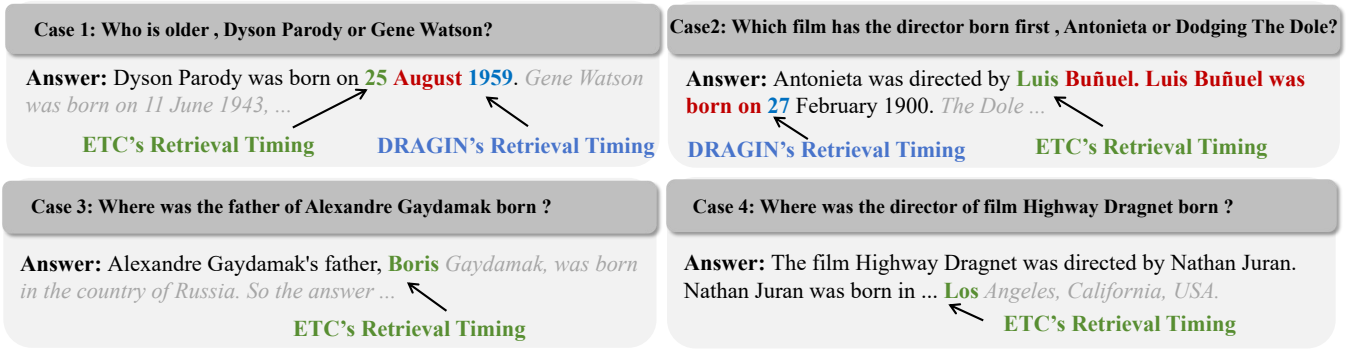

Figure 4: Illustrative cases of delayed retrieval. The first two cases demonstrate delayed retrieval, where green tokens indicate ETC's retrieval timing, blue tokens represent DRAGIN's retrieval timing, and red tokens highlight incorrectly generated tokens caused by delayed retrieval. The last two cases illustrate missing retrieval, which is a special case of delayed retrieval.

are redundant and ineffective. Specifically, in 54 samples where DRAGIN retrieves at the first token, only 10 result in improved answers, while the remaining 44 are equal to or worse than those without retrieval, yielding a redundancy rate of 81.5%. This observation is consistent with the high redundant retrieval ratio reported in Table 6.

In contrast, ETC not only triggers retrieval at earlier average positions but also achieves more effective retrieval by leveraging meaningful uncertainty trends. These findings confirm ETC's advantage in both timing precision and retrieval effectiveness.

6.5 Case Study

In addition to the quantitative analysis, we present several intuitive cases from 2WikiMultihopQA to illustrate how different dynamic RAG methods behave during the generation process, as shown in Figure 4.

The first two cases in Figure 4 show that DRAGIN generates multiple incorrect tokens before triggering retrieval, exhibiting the delayed retrieval issue. This occurs because DRAGIN determines retrieval timing based solely on single-token confidence, which may fail to trigger retrieval early enough when external knowledge is required. In contrast, ETC leverages entropy change trends to intervene at the right moment, leading to more coherent and accurate generation with timely access to external knowledge.

We also identify a special form of delayed retrieval, referred to as missing retrieval, where the dynamic RAG system fails to trigger retrieval at all during the generation process. The last two cases in Figure 4 show that DRAGIN fails to detect the appropriate retrieval timing, whereas ETC intervenes effectively, resulting in the correct answer.

7 Related Work

Retrieval augmented generation is an efficient and effective approach to help LLMs obtaining necessary external knowledge(Fan et al. 2024). Existing works mainly focusing on training-based RAG system (Yoran et al. 2023; Luo et al. 2024; Fang et al. 2024; Xu et al. 2024) and training-free RAG system (Izacard and Grave 2020; Wang et al. 2023; Jiang et al. 2023; Su et al. 2024b). This paper focuses on the later one since it is more lightweight and efficient in practical scenarios.

In the era of LLM, early research mainly explores designing more suitable prompts for high-quality retrieved text (Shi et al. 2023; Wang, Yang, and Wei 2023; Yu et al. 2023),these methods typically conduct retrieval operations only once at the start of the generation process. Lately, people found that not all retrieval operations are beneficial for LLM's generation, improper or redundant augmented information may cause negative influence on the performance (Wang et al. 2023; Ni et al. 2024; Su et al. 2024a). Based on the above observation, more research explore to active the retrieval operation when LLM needed, named as dynamic RAG. Borgeaud et al. (2022); Trivedi et al. (2022); Ram et al. (2023) proposed to retrieve every n tokens or every sentence, making LLM receive new knowledge during generation process. While Jiang et al. (2023); Wang et al. (2024b); Tao et al. (2024) propose to determine the retrieval timing based on the prediction confidence of the generated token or the internal states. Su et al. (2024b) further considers the importance of each token to find more reasonable retrieval timing.

8 Conclusion

In this paper, we introduced Entropy-Trend Constraint (ETC), a training-free method for selecting optimal retrieval timing in dynamic retrieval-augmented generation. Unlike prior approaches that rely on a single token's confidence, ETC models entropy trends over time to detect rising uncertainty and trigger retrieval more effectively. Extensive experiments on six QA datasets across multiple LLMs demonstrate that ETC consistently outperforms strong baselines while reducing retrieval frequency. Beyond performance improvements, our findings shed light on the role of temporal uncertainty modeling in retrieval-aware generation, offering practical guidance for designing future dynamic RAG systems.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62276089 and Grant No. 62506115), the Natural Science Foundation of Tianjin (Grant No. 24JCJQJC00200 and Grant No. 24JCQNJC01230), the Natural Science Foundation of Hebei Province (Grant No. F2024202064 and Grant No. F2025202020), the Science Research Project of Hebei Education Department (Grant No. BJ2025004), the Ministry of Human Resources and Social Security of China (Grant No. RSTH-2023-135-1), and the Science and Technology Program of Hebei Province (Grant No. 24464401D).

References

Ames, W. F. 2014. *Numerical methods for partial differential equations*. Academic press.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G. B.; Lespiau, J.-B.; Damoc, B.; Clark, A.; et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, 2206–2240. PMLR.

Chen, J.; Lin, H.; Han, X.; and Sun, L. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17754–17762.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. *URL https://lmsys. org/blog/2023-03-30-vicuna*, 3(5).

Fadeeva, E.; Rubashevskii, A.; Shelmanov, A.; Petrakov, S.; Li, H.; Mubarak, H.; Tsymbalov, E.; Kuzmin, G.; Panchenko, A.; Baldwin, T.; Nakov, P.; and Panov, M. 2024. Fact-Checking the Output of Large Language Models via Token-Level Uncertainty Quantification. *ArXiv*, abs/2403.04696.

Fan, W.; Ding, Y.; Ning, L.; Wang, S.; Li, H.; Yin, D.; Chua, T.-S.; and Li, Q. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 6491–6501.

Fang, F.; Bai, Y.; Ni, S.; Yang, M.; Chen, X.; and Xu, R. 2024. Enhancing Noise Robustness of Retrieval-Augmented Language Models with Adaptive Adversarial Training. In *Annual Meeting of the Association for Computational Linguistics*.

Farquhar, S.; Kossen, J.; Kuhn, L.; and Gal, Y. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630: 625 – 630.

Ferguson, J.; Gardner, M.; Hajishirzi, H.; Khot, T.; and Dasigi, P. 2020. IIRC: A dataset of incomplete information reading comprehension questions. *arXiv preprint arXiv:2011.07127*.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; and Wang, H. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; and Berant, J. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9: 346–361.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ho, X.; Nguyen, A.-K. D.; Sugawara, S.; and Aizawa, A. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Izacard, G.; and Grave, E. 2020. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. *ArXiv*, abs/2007.01282.

Jiang, Z.; Xu, F. F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W. W.; and Lu, X. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. *ArXiv*, abs/1909.06146.

Jordán, K. 1965. *Calculus of finite differences*, volume 33. American Mathematical Soc.

Kandpal, N.; Deng, H.; Roberts, A.; Wallace, E.; and Raffel, C. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, 15696–15707. PMLR.

Levy, H.; and Lessman, F. 1992. *Finite difference equations*. Courier Corporation.

Li, J.; Sun, S.; Yuan, W.; Fan, R.-Z.; Zhao, H.; and Liu, P. 2023. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.

Li, J.; Yuan, Y.; and Zhang, Z. 2024. Enhancing llm factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. *arXiv preprint arXiv:2403.10446*.

Li, Z.; Xu, X.; Shen, T.; Xu, C.; Gu, J.-C.; Lai, Y.; Tao, C.; and Ma, S. 2024. Leveraging large language models for nlg evaluation: Advances and challenges. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 16028–16045.

Luo, K.; Liu, Z.; Xiao, S.; Zhou, T.; Chen, Y.; Zhao, J.; and Liu, K. 2024. Landmark Embedding: A Chunking-Free Embedding Method For Retrieval Augmented Long-Context Large Language Models. In *Annual Meeting of the Association for Computational Linguistics*.

Maekawa, S.; Iso, H.; Gurajada, S.; and Bhutani, N. 2024. Retrieval Helps or Hurts? A Deeper Dive into the Efficacy of Retrieval Augmentation to Language Models. *arXiv preprint arXiv:2402.13492*.

Melz, E. 2023. Enhancing llm intelligence with arm-rag: Auxiliary rationale memory for retrieval augmented generation. *arXiv preprint arXiv:2311.04177*.

Mousavi, S. M.; Alghisi, S.; and Riccardi, G. 2024. Is Your LLM Outdated? Benchmarking LLMs & Alignment

Algorithms for Time-Sensitive Knowledge. *arXiv preprint arXiv:2404.08700*.

Ni, S.; Bi, K.; Guo, J.; and Cheng, X. 2024. When Do LLMs Need Retrieval Augmentation? Mitigating LLMs' Overconfidence Helps Retrieval Augmentation. In *Annual Meeting of the Association for Computational Linguistics*.

Nikitin, A. V.; Kossen, J.; Gal, Y.; and Marttinen, P. 2024. Kernel Language Entropy: Fine-grained Uncertainty Quantification for LLMs from Semantic Similarities. *ArXiv*, abs/2405.20003.

Ram, O.; Levine, Y.; Dalmedigos, I.; Muhlgay, D.; Shashua, A.; Leyton-Brown, K.; and Shoham, Y. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11: 1316–1331.

Ren, R.; Wang, Y.; Qu, Y.; Zhao, W. X.; Liu, J.; Tian, H.; Wu, H.; Wen, J.-R.; and Wang, H. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.

Shi, W.; Min, S.; Yasunaga, M.; Seo, M.; James, R.; Lewis, M.; Zettlemoyer, L.; and tau Yih, W. 2023. REPLUG: Retrieval-Augmented Black-Box Language Models. In *North American Chapter of the Association for Computational Linguistics*.

Su, H.; Yen, H.; Xia, M.; Shi, W.; Muennighoff, N.; yu Wang, H.; Liu, H.; Shi, Q.; Siegel, Z. S.; Tang, M.; Sun, R.; Yoon, J.; Arik, S. Ö.; Chen, D.; and Yu, T. 2024a. BRIGHT: A Realistic and Challenging Benchmark for Reasoning-Intensive Retrieval. *ArXiv*, abs/2407.12883.

Su, W.; Tang, Y.; Ai, Q.; Wu, Z.; and Liu, Y. 2024b. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. *arXiv preprint arXiv:2403.10081*.

Tao, S.; Yao, L.; Ding, H.; Xie, Y.; Cao, Q.; Sun, F.; Gao, J.; Shen, H.; and Ding, B. 2024. When to Trust LLMs: Aligning Confidence with Response Quality. *ArXiv*, abs/2404.17287.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Tsatsaronis, G.; Balikas, G.; Malakasiotis, P.; Partalas, I.; Zschunke, M.; Alvers, M. R.; Weissenborn, D.; Krithara, A.; Petridis, S.; Polychronopoulos, D.; et al. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16: 1–28.

Wang, C.; Long, Q.; Xiao, M.; Cai, X.; Wu, C.; Meng, Z.; Wang, X.; and Zhou, Y. 2024a. Biorag: A rag-llm framework for biological question reasoning. *arXiv preprint arXiv:2408.01107*.

Wang, H.; Xue, B.; Zhou, B.; Zhang, T.; Wang, C.; Chen, G.; Wang, H.; and Wong, K.-F. 2024b. Self-DC: When to Reason and When to Act? Self Divide-and-Conquer for Compositional Unknown Questions.

Wang, L.; Yang, N.; and Wei, F. 2023. Query2doc: Query Expansion with Large Language Models. In *Conference on Empirical Methods in Natural Language Processing*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wang, Y.; Li, P.; Sun, M.; and Liu, Y. 2023. Self-Knowledge Guided Retrieval Augmentation for Large Language Models. In *Conference on Empirical Methods in Natural Language Processing*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Xiong, H.; Bian, J.; Li, Y.; Li, X.; Du, M.; Wang, S.; Yin, D.; and Helal, S. 2024. When search engine services meet large language models: visions and challenges. *IEEE Transactions on Services Computing*.

Xu, S.; Pang, L.; Yu, M.; Meng, F.; Shen, H.; Cheng, X.; and Zhou, J. 2024. Unsupervised Information Refinement Training of Large Language Models for Retrieval-Augmented Generation. *ArXiv*, abs/2402.18150.

Yang, Y.; Zhang, Y.; Hu, Y.; Guo, Y.; Gan, R.; He, Y.; Lei, M.; Zhang, X.; Wang, H.; Xie, Q.; et al. 2024. Ucfe: A user-centric financial expertise benchmark for large language models. *arXiv preprint arXiv:2410.14059*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yoran, O.; Wolfson, T.; Ram, O.; and Berant, J. 2023. Making Retrieval-Augmented Language Models Robust to Irrelevant Context. *ArXiv*, abs/2310.01558.

Yu, W.; Zhang, Z.; Liang, Z.; Jiang, M.; and Sabharwal, A. 2023. Improving Language Models via Plug-and-Play Retrieval Feedback. *ArXiv*, abs/2305.14002.