Adaptive Retrieval without Self-Knowledge? Bringing Uncertainty Back Home

Viktor Moskvoretskii[1,3], Maria Marina[2,1], Mikhail Salnikov[2,1], Nikolay Ivanov[1], Sergey Pletenev[2,1], Daria Galimzianova[4], Nikita Krayko[4], Vasily Konovalov[2,5], Irina Nikishina[6], and Alexander Panchenko[1,2]

[1]Skoltech, [2]AIRI, [3]HSE University, [4]MTS AI, [5]MIPT, [6]University of Hamburg

Correspondence: {V.Moskvoretskii, A.Panchenko}@skol.tech

Abstract

Retrieval Augmented Generation (RAG) improves correctness of Question Answering (QA) and addresses hallucinations in Large Language Models (LLMs), yet greatly increase computational costs. Besides, RAG is not always needed as may introduce irrelevant information. Recent adaptive retrieval methods integrate LLMs' intrinsic knowledge with external information appealing to LLM self-knowledge, but they often neglect efficiency evaluations and comparisons with uncertainty estimation techniques. We bridge this gap by conducting a comprehensive analysis of 35 adaptive retrieval methods, including 8 recent approaches and 27 uncertainty estimation techniques, across 6 datasets using 10 metrics for QA performance, self-knowledge, and efficiency. Our findings show that uncertainty estimation techniques often outperform complex pipelines in terms of efficiency and self-knowledge, while maintaining comparable QA performance.

1 Introduction

Large Language Models have gained increased popularity due to their remarkable performance across diverse tasks, such as question answering (QA) (Yang et al., 2018; Kwiatkowski et al., 2019). At the same time, hallucinations represent a substantial challenge for LLMs. Solely utilizing only parametric knowledge in generating trustworthy content is limited by the knowledge boundaries of LLMs (Yin et al., 2024), which may potentially lead to internal hallucinations (Ding et al., 2024). While external information via RAG (Lewis et al., 2020b) can potentially help to fill these gaps, it raises the possibility of irrelevance, thus leading to the error accumulation (Shi et al., 2023) and increasing the likelihood of external hallucinations (Ding et al., 2024).

To balance between the intrinsic knowledge of LLMs and external information, adaptive retrieval methods have emerged (Su et al., 2024b; Ding et al., 2024; Jeong et al., 2024). These methods rely on LLM **self-knowledge** — model capacity to recognize its own knowledge (Yin et al., 2023) — and determine when it lacks critical information.

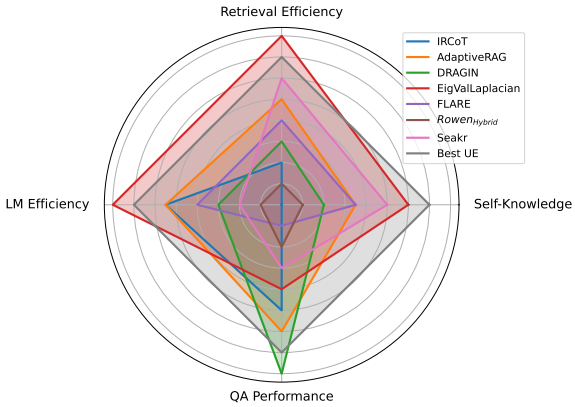

Figure 1: Performance comparison of the state-of-the art models across efficiency metrics (number of LLM calls, Retrieval calls), QA quality metric (In-Accuracy), and the ability to identify self-knowledge (ROCAUC). The plot demonstrates the reverted ranks of the methods across 6 datasets.

Adaptive retrieval methods may not only improve answer correctness, but also substantially decrease retrieval calls, enhancing efficiency. While recent methods have focused extensively on the retrieval calls (Su et al., 2024b; Jeong et al., 2024; Trivedi et al., 2023), they often overlook the cost of LLM calls, which can be even more expensive, especially with proprietary models. Furthermore, recent studies of complex pipelines do not assess self-knowledge abilities and lack comparisons with well-established uncertainty estimation methods, such as Mean Entropy (Fomicheva et al., 2020).

To address these limitations, we conduct a comprehensive study of 35 adaptive retrieval systems, including 8 recently published methods and 27 established uncertainty estimation methods, across 6 QA datasets covering both simple one-hop and

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6355–6384
July 27 - August 1, 2025

complex multi-hop questions. We evaluate these methods in terms of the QA performance, self-knowledge, and two types of efficiency, using a total of 10 metrics. Our evaluation, shown in Figure 1, reveals that no single method dominates across all axes. However, well-established uncertainty estimation methods are often more useful compared to recently published, more complex pipelines.

Finally, we provide a rigorous in-depth assessment of the out-of-distribution (OOD) performance of uncertainty methods and analyze the complexity of their functional classes.

Our contributions and findings are as follows:

1. A consistent study of 35 adaptive retrieval methods on 6 single- and multi-hop datasets, evaluating QA performance, self-knowledge, and efficiency across 10 metrics.
2. The first comprehensive application and comparison of 27 well-established uncertainty estimation methods for adaptive retrieval, showcasing their potential and efficiency.
3. An in-depth analysis of uncertainty methods for adaptive retrieval, covering OOD transfer and examining the complexity of their functional classes.

We make data and all models publicly available.[1]

2 Related Work

Retrieval-Augmented Generation methods are widely used to enhance the performance of LLMs in many tasks, like up-to-date information (Jiang et al., 2024) or questions about rare entities in which LLM shows poor generation quality due to lack of inner knowledge (Allen-Zhu and Li, 2024). In the simplest case, the input sequence of the question is used as a query for databases or search engines. The resulting information is then incorporated as an additional context, proven effective for a variety of tasks (Khandelwal et al., 2020; Lewis et al., 2020a) and models (Borgeaud et al., 2022; Ram et al., 2023; Socher et al., 2013). All these methods are applied to the retrieval once before generation, so they are often combined under the name single-round retrieval augmentation.

Adaptive Retrieval-Augmented Generation methods perform retrieval for every query may be both inefficient and unnecessary. Moreover, retrieving knowledge at every step may be misleading or even conflicting with LLM's parameters (Simhi et al., 2024). Adaptive retrieval methods have emerged as an attempt to understand whether LLM needs external knowledge by exploiting models' self-knowledge abilities.

The decision to retrieve may depend on different criteria. It may be based on the text outputs of LLMs (Trivedi et al., 2023) or text consistency (Ding et al., 2024), on the self-aware uncertainty of LLMs from their internal states (Jiang et al., 2023; Su et al., 2024b; Yao et al., 2024) or using a trained classifier to decide whether to retrieve (Jeong et al., 2024). While early work performed retrieval at fixed points or per step (Trivedi et al., 2023; Press et al., 2022), recent research explores more interactive and globally informed retrieval strategies that coordinate multiple reasoning steps and adjust retrieval based on verification signals (Xu et al., 2024).

Uncertainty Estimation (UE) measures the confidence in LLM predictions and can be classified into white-box and black-box methods. White-box methods require access to internal model details, such as logits or layer outputs, and are divided into information-based (using token or sequence probabilities from a single model), ensemble-based (leveraging probabilities from different model versions), and density-based (constructing a probability density from latent representations). Black-box methods, in contrast, only require access to the model's output (Fadeeva et al., 2023).

3 Methods

In this section, we briefly introduce the existing adaptive retrieval methods. More details can be found in Appendix F.

3.1 End-to-End Methods

IRCoT (**I**nterleaving **R**etrieval in a **CoT**) is a dynamic approach that adds extra relevant passages from the retriever to the context if the current CoT step has not produced the answer yet. The query for extra context is based on the last generated CoT sentence (Trivedi et al., 2023).

Adaptive RAG (Jeong et al., 2024) uses the classifier based on the T5-large model (Raffel et al., 2020) that predicts one of the three outcomes: whether not to retrieve at all, retrieve once and retrieve multiple times with IRCoT.

FLARE (**F**orward-**L**ooking **A**ctive **Re**trieval augmented generation) is a method that retrieves

[1] https://github.com/s-nlp/AdaRAGUE

context when token probability falls below a threshold, regenerating the response until the next uncertain token or completion (Jiang et al., 2023).

DRAGIN (**D**ynamic **R**etrieval **A**ugmented **G**eneration based on **I**nformation **N**eeds) monitors token probabilities like FLARE but filters stopwords to identify uncertainty tokens. It improves context retrieval by reformulating queries using attention weights and reasoning (Su et al., 2024b).

Rowen (**R**etrieve **O**nly **W**hen It **N**eeds) is a consistency-based approach with two components: the Consistency Language, which measures answer consistency across English and Chinese, and the Consistency Model, which evaluates semantic coherence across models. Both output inconsistency scores to trigger retrieval. The Rowen Hybrid combines both components (Ding et al., 2024).

SeaKR (**Se**lf-**a**ware **K**nowledge **R**etrieval) uses an Uncertainty Module (UM) to monitor LLM internal states and trigger retrieval when uncertainty exceeds a threshold. A re-ranking component selects a snippet that reduces uncertainty and improves factual accuracy (Yao et al., 2024).

3.2 Uncertainty Estimation Methods

For uncertainty estimation, we employ 27 different methods, described in detail in Table 15. In the main part of our paper, we focus on the 5 best-performing uncertainty estimation methods, which include approaches from various method families:

- **Lexical Similarity**: Measures a consistency score based on the average similarity of sampled responses (Fomicheva et al., 2020).
- **Max Entropy**: Computes the entropy of each token and aggregates it for the sequence using the maximum value (Fomicheva et al., 2020).
- **Mean Entropy**: Computes the entropy of each token and aggregates it for the sequence using the mean value (Fomicheva et al., 2020).
- **SAR**: Measures the entropy of each token, reweights it based on token relevance, and aggregates the values using a sum over the sequence (Duan et al., 2023).
- **EigValLaplacian**: Computes the sum of Laplacian eigenvalues by constructing a weighted graph based on the consistency of sampled responses (Lin et al., 2023).

4 Experimental Setup

In this section, we briefly discuss the implementation details and the evaluation setup.

4.1 Implementation Details

We use the LLaMA 3.1-8b-instruct model (Dubey et al., 2024) with the default generation parameters for all experiments. The baseline methods follow their original protocols, including prompting and parameter settings, while uncertainty estimation methods use the AdaptiveRAG protocol (Jeong et al., 2024), with the same prompt and few-shot examples.

For all methods, we use the BM25 retriever (Robertson et al., 1994) with Elasticsearch 7.17.9[2] and the Wikipedia corpus preprocessed by Karpukhin et al. (2020), following previous studies (Su et al., 2024a; Yao et al., 2024).

Uncertainty method scores are computed on both training and test sets using the LM-Polygraph (Fadeeva et al., 2023). A set of classifiers are trained on the training set scores, with the best classifier's performance reported based on downstream metrics. Additional details are provided in Appendix E.

4.2 Datasets

We use the single-hop and multi-hop QA datasets in the same experimental setup to replicate a real-world scenario where various queries have different difficulties. The choice of datasets is standard for the task with the single-hop questions – SQUAD v1.1 (Rajpurkar et al., 2016), Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and the datasets with the complex ones – MuSiQue (Trivedi et al., 2022), HotpotQA (Yang et al., 2018), and 2WikiMultiHopQA (Ho et al., 2020), following previous papers (Trivedi et al., 2023; Jeong et al., 2024; Su et al., 2024b; Yao et al., 2024). To ensure consistency, we use the subsets of 500 questions of the original test parts of the datasets from previous studies (Trivedi et al., 2023; Jeong et al., 2024) .

4.3 Evaluation

We conduct a comprehensive evaluation using QA downstream metrics, efficiency metrics, and self-knowledge metrics to broadly cover every aspect of the model. To fairly compare performance across datasets, we also use methods ranks on each dataset

[2] https://www.elastic.co/elasticsearch

(smaller rank indicates better performance) and average the ranks. This ensures a balanced evaluation, as performance gains may vary in significance across datasets.

4.3.1 Downstream QA Metrics

To assess the final QA system quality we use In-Accuracy, EM and F1, following previous studies (Mallen et al., 2023; Baek et al., 2023; Asai et al., 2024; Jeong et al., 2024), where:

- **In-Accuracy (InAcc)** evaluates whether the predicted answer includes the ground truth.
- **Exact Match (EM)** measures the exact match of prediction with the ground truth.
- **F1** quantifies the degree of token overlap between the predicted answer and the ground truth answer.

We primarily rely on In-Accuracy as the main metric, as it is more robust to answer variations compared to EM and provides a better measure of correctness than F1. Additionally, the overall trends across these metrics are generally consistent.

4.3.2 Efficiency Metrics

In addition to enhanced quality, adaptive retrieval procedures must also demonstrate improvements in efficiency; otherwise, consistent retrieval might remain superior. To evaluate it, we measure:

- **Retriever Calls (RC)**: The average number of retriever calls made by the system to answer a single question, following Jeong et al. (2024).
- **LM Calls (LMC)**: The average number of calls to the Language Model per question. Some systems may invoke the LM multiple times to calculate uncertainty, rephrase questions or generate additional rationales.

4.3.3 Self-Knowledge Metrics

Self-knowledge is defined as a model's ability to recognize its own knowledge (Yin et al., 2023). Measuring self-knowledge provides insight into the effectiveness of a method's adaptive retrieval component, as downstream performance is often influenced by external factors such as retriever selection, language model generation parameters, etc.

The task of identifying self-knowledge is formulated as a binary classification problem, where the ground truth label y is derived from the In-Accuracy of the model's response without external knowledge. Each method f can be represented as a function mapping input text x to a real-valued self-knowledge score $f(x) \in \mathbb{R}$, where higher values indicate higher self-knowledge. The classification task is then performed by a classifier C, producing the final prediction $\hat{y} = C(f(x)) \in \{0, 1\}$.

For evaluation, we adopt metrics established in prior uncertainty estimation research (Fadeeva et al., 2024b; Tao et al., 2024) and reflexive self-knowledge analysis (Ni et al., 2024).

- **ROC-AUC (AUC)** evaluates the robustness of the method's self-knowledge identification performance: $AUC(f(\mathbf{x}), \mathbf{y})$.
- **Spearman Correlation (Corr)** measures the alignment between the self-knowledge scores and the ground truth: $Corr(f(\mathbf{x}), \mathbf{y})$.
- **Accuracy** quantifies the correctness of self-knowledge classifier: $\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\hat{y}_i, y_i\}$.
- **Overconfidence** is a fraction of incorrect answers where the method was confident about self-knowledge reflecting how often the method incorrectly assumes that the model possesses the required knowledge when it does not: $\frac{1}{\sum_i (1-\hat{y}_i)} \sum_i (1 - \mathbb{1}\{\hat{y}_i, y_i\}) \cdot (1 - \hat{y}_i)$.
- **Underconfidence** is a fraction of correct answers where the method was unconfident about self-knowledge reflecting how often the method fails to recognize that the model already has the required knowledge: $\frac{1}{\sum_i \hat{y}_i} \sum_i (1 - \mathbb{1}\{\hat{y}_i, y_i\}) \cdot \hat{y}_i$.

5 Results

In the following sections, we describe the results of baseline and uncertainty methods for downstream performance, efficiency and self-knowledge. Along with the end-to-end and UE methods, we also apply two additional methods for better comparison. "Best UE" refers to the top-performing uncertainty estimation method for each dataset. "Ideal" represents the performance of a system with an oracle providing ideal predictions for the need to retrieve.

5.1 Downstream and Efficiency Performance

The results in Table 1 show that uncertainty estimation methods outperform baseline methods on single-hop datasets and perform comparably on multi-hop datasets, while being significantly more compute-efficient, often several times cheaper.

Method	NQ			SQUAD			TQA			2Wiki			HotPot			Musique		
	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓
Never RAG	0.446	1.0	0.00	0.176	1.0	0.00	0.636	1.0	0.00	0.318	1.0	0.00	0.286	1.0	0.00	0.106	1.0	0.00
Always RAG	0.496	1.0	1.00	0.312	1.0	1.00	0.610	1.0	1.00	0.374	1.0	1.00	0.410	1.0	1.00	0.100	1.0	1.00
IRCoT	0.478	2.7	2.70	0.268	2.7	2.68	0.608	2.7	2.74	0.454	4.4	4.38	**0.438**	3.5	3.45	0.138	4.1	4.08
AdaptiveRAG	0.496	2.0	0.98	0.286	2.0	0.97	0.628	1.5	0.54	0.454	5.2	2.64	0.414	4.6	2.34	**0.140**	3.6	3.63
DRAGIN	0.480	4.5	2.24	0.298	4.3	2.14	**0.666**	4.1	2.06	**0.456**	5.8	2.92	0.430	5.1	2.56	0.134	6.3	3.15
FLARE	0.450	3.1	2.07	0.238	3.1	2.08	0.648	2.1	1.39	0.424	3.9	2.85	0.372	5.1	4.07	0.090	4.1	3.10
Rowen_{CL}	0.494	29.5	7.24	0.196	29.2	7.19	0.656	28.7	7.06	0.444	32.9	7.87	0.354	31.9	7.67	0.104	42.1	9.52
Rowen_{CM}	0.494	29.5	7.27	0.196	29.2	7.20	0.656	28.7	7.12	0.444	32.9	7.87	0.356	31.9	7.70	0.104	42.1	9.52
$\text{Rowen}_{\text{Hybrid}}$	0.494	55.0	7.27	0.196	54.3	7.15	0.656	53.4	6.93	0.444	61.8	7.85	0.354	59.8	7.63	0.102	80.2	9.48
Seakr	0.406	14.6	1.00	0.268	14.6	1.00	0.656	14.6	1.00	0.398	12.3	2.44	0.424	9.9	1.76	0.118	12.3	2.40
EigValLaplacian	**0.512**	1.8	0.81	0.314	2.0	1.00	0.640	1.3	0.26	**0.384**	2.0	0.98	0.410	1.9	0.91	**0.102**	2.0	**0.99**
Lex-Similarity	**0.512**	1.6	**0.58**	**0.318**	2.0	**0.96**	0.646	1.2	0.22	0.376	2.0	0.97	0.410	2.0	0.95	0.100	2.0	1.00
Max Entropy	0.506	1.7	0.73	0.312	2.0	1.00	0.650	1.2	**0.22**	0.376	2.0	0.95	**0.414**	2.0	0.99	0.100	2.0	1.00
Mean Entropy	0.498	1.9	0.88	0.314	2.0	**0.95**	0.650	1.3	0.30	0.378	1.9	**0.93**	0.410	2.0	0.99	0.100	2.0	1.00
SAR	0.500	1.8	0.79	0.312	2.0	1.00	0.642	1.3	0.29	0.380	2.0	0.97	0.412	1.9	**0.90**	0.100	2.0	1.00
Best UE	**0.512**	1.8	0.81	**0.318**	2.0	0.96	**0.662**	1.3	0.28	**0.384**	2.0	0.98	**0.414**	2.0	0.99	**0.104**	2.0	**0.99**
Ideal	0.608	1.6	0.55	0.360	1.8	0.82	0.736	1.4	0.36	0.500	1.7	0.68	0.460	1.7	0.71	0.164	1.9	0.89

Table 1: QA Performance of adaptive retrieval and uncertainty methods. 'Best UE' refers to the top-performing uncertainty estimation method for each dataset. 'Ideal' represents the performance of a system with an oracle providing ideal predictions for the need to retrieve. 'InAcc' denotes In-Accuracy, measuring the QA system's performance. 'LMC' indicates the mean number of LM calls per question, and 'RC' represents the mean number of retrieval calls per question.

While baseline methods may achieve slightly better performance on some datasets, they require multiple calls to both the language model and retriever, leading to higher computational costs. In contrast, uncertainty estimation methods consistently require fewer than one retriever call and two or less LM calls per question, significantly reducing inference costs.

Uncertainty estimation for adaptive retrieval consistently outperforms constant retrieval in terms of performance and efficiency. However, analysis of the Ideal uncertainty estimator reveals that current methods still fall short of perfect performance, both in terms of efficiency and In-Accuracy, highlighting the ongoing challenge of accurately identifying self-knowledge within the model.

Takeaway 1: Uncertainty methods outperform baselines on single-hop tasks, match them on multi-hop tasks, and are far more efficient. The "Ideal" estimator highlights room for improvement in the self-knowledge identification.

5.2 Self-Knowledge Performance

The results in Table 2 demonstrate that, despite strong downstream performance, most adaptive retrieval methods may lack the ability to accurately identify self-knowledge, exhibiting near-zero correlation and random predictions. For instance, while DRAGIN typically dominates on downstream tasks, it performs poorly on self-knowledge metrics.

In contrast, SeaKR exhibits strong self-knowledge identification on single-hop datasets, underscoring the value of inspecting the internal states of language models. However, SeaKR's performance declines on multi-hop datasets, where internal states may provide limited information about the model's knowledge of more complex questions. For multi-hop tasks, AdaptiveRAG demonstrates superior results, highlighting the effectiveness of reflexive trainable methods, which apparently handle complex reasoning better.

These results suggest that internal-state uncertainty excels for simple questions, while reflexive uncertainty methods are better suited for complex reasoning tasks.

According to the results in Figure 2, nearly all baseline models, except for AdaptiveRAG, exhibit a tendency to either consistently overestimate self-knowledge or, conversely, to be underconfident. In contrast, uncertainty methods strike the best balance between overconfidence and underconfidence, demonstrating more adequate and reliable values.

Overall, uncertainty estimation methods consistently exhibit the strongest ability to identify self-knowledge, ranking first or second across all methods. These findings emphasize the need for a more thorough evaluation of adaptive retrieval methods, beyond relying solely on downstream performance, showing no significant correlation, further shown in Table 14.

Takeaway 2: Internal-based SeaKR excels at simple tasks, while reflexive AdaptiveRAG performs better on complex ones. Uncertainty methods provide the most reliable self-knowledge estimates, emphasizing evaluation beyond QA performance.

Method	NQ			SQUAD			TQA			2Wiki			HotPot			Musique		
	Acc	Corr	AUC	Acc	Corr	AUC	Acc	Corr	AUC	Acc	Corr	AUC	Acc	Corr	AUC	Acc	Corr	AUC
AdaptiveRAG	0.57	0.06	0.54	0.73	0.10	0.58	0.51	-0.02	0.49	**0.72**	**0.34**	**0.71**	0.71	0.19	0.62	0.88	0.15	0.64
DRAGIN	0.55	0.12	0.57	0.82	0.11	0.58	0.36	0.03	0.52	0.68	-0.07	0.46	0.71	0.01	0.51	0.89	-0.01	0.49
FLARE	0.59	0.16	0.59	0.54	0.11	0.58	0.58	0.12	0.57	0.51	0.20	0.62	0.42	0.06	0.54	0.59	0.01	0.51
Rowen_{CL}	0.45	-0.14	0.44	0.18	-0.06	0.47	0.64	-0.07	0.47	0.32	-0.10	0.46	0.29	-0.13	0.44	0.11	0.00	0.50
Rowen_{CM}	0.45	-0.03	0.49	0.18	-0.06	0.47	0.64	-0.13	0.44	0.32	0.02	0.51	0.29	-0.14	0.44	0.11	-0.02	0.49
$\text{Rowen}_{\text{Hybrid}}$	0.45	-0.12	0.44	0.17	-0.07	0.46	0.63	-0.13	0.43	0.32	-0.04	0.48	0.29	-0.17	0.41	0.11	-0.01	0.49
Seakr	0.55	**0.24**	**0.64**	0.82	**0.36**	**0.77**	0.36	**0.47**	**0.78**	0.68	-0.22	0.37	0.71	0.08	0.55	0.89	0.06	0.56
EigValLaplacian	0.60	0.17	0.60	0.83	0.10	0.57	0.70	0.34	0.71	0.69	0.19	0.62	**0.73**	0.27	0.67	0.89	0.12	0.62
Lex-Similarity	0.61	0.22	0.63	**0.84**	0.22	0.67	**0.73**	0.39	0.74	0.68	0.21	0.63	**0.73**	0.30	0.69	**0.90**	0.08	0.59
Max Entropy	**0.63**	0.20	0.62	0.82	0.25	0.69	0.72	0.35	0.71	0.69	0.19	0.62	**0.73**	0.29	0.69	0.89	**0.18**	**0.67**
Mean Entropy	0.61	0.20	0.62	**0.84**	0.32	0.74	0.72	0.36	0.72	0.68	0.28	0.68	0.72	**0.31**	**0.70**	**0.90**	0.15	0.64
SAR	0.61	0.23	0.63	0.83	0.28	0.71	0.72	0.38	0.73	0.69	0.23	0.65	**0.73**	0.30	0.69	0.89	0.17	0.66

Table 2: **Self-knowledge** metrics for adaptive retrieval and uncertainty methods. 'Acc' and 'AUC' refer to accuracy and ROC-AUC, respectively, for identifying self-knowledge. 'Corr' denotes the Spearman correlation with the self-knowledge label. Bold values indicate the highest score, underlined values represent the second-highest score.

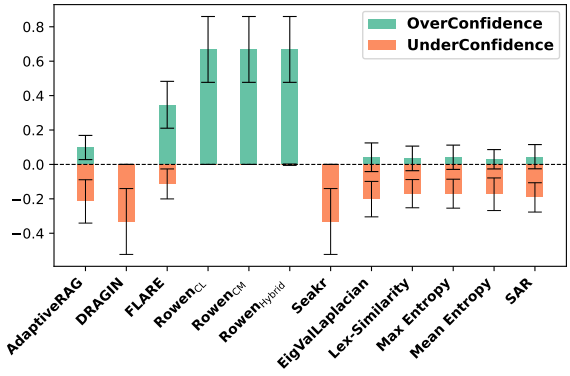

Figure 2: Average overconfidence and underconfidence for each method. Deviation from the zero value is undesirable and indicates erroneous behavior. High **OverConfidence** values reflect cases where the method incorrectly assumes the model has the required knowledge when it does not. High **UnderConfidence** values indicate instances where the method fails to recognize that the model already possesses the required knowledge.

5.3 Uncertainty Estimation

We analyze 27 uncertainty estimation methods across QA performance, efficiency, and self-knowledge, categorizing them by underlying approach. Methods are ranked based on their average performance across datasets, with smaller ranks indicating better results.

As shown in Figure 3, EigValLaplacian and Lex-Similarity rank highest for In-Accuracy, while SAR variants and Mean Entropy dominate for ROC-AUC, highlighting an inconsistency between self-knowledge and downstream performance. This discrepancy is further evidenced by a moderate Spearman correlation of 0.65 between In-Accuracy and ROC-AUC ranks, likely due to differing sensitivities to Type I and II errors. EigValLaplacian also ranks highest for Retrieval Calls, indicating overconfidence.

For our main analysis, we select uncertainty methods with the best QA performance: EigValLaplacian, Lex-Similarity, and Max Entropy and top self-knowledge methods: SAR and Mean Entropy for self-knowledge assessment. Internal-state methods generally rank lower for In-Accuracy and ROC-AUC but perform better in efficiency, suggesting overconfidence. Consistency-based methods excel in QA performance but drop in self-knowledge, lagging behind logit-based methods, indicating better stability to distribution shifts.

The Hybrid method balances all metrics, ranking in the top-5 for In-Accuracy and ROC-AUC and first for efficiency. However, it requires calculating all uncertainty estimates, introducing computational overhead, which may still be justified in retrieval-limited scenarios.

Finally, we analyze feature importance for the Hybrid method in details in Figure 16 in Appendix D.

> **Takeaway 3:** Consistency-based methods excel in downstream performance but lag in self-knowledge, while logit-based methods dominate self-knowledge metrics. The Hybrid method balances all metrics but incurs higher computational costs.

6 Out-of-Domain Transfer

To analyze the robustness of UE methods on out-of-domain (OOD) datasets, we evaluate their performance across all possible dataset pairs by training on each dataset and testing on every other. The relative change in performance, expressed as a percentage compared to in-domain performance (see

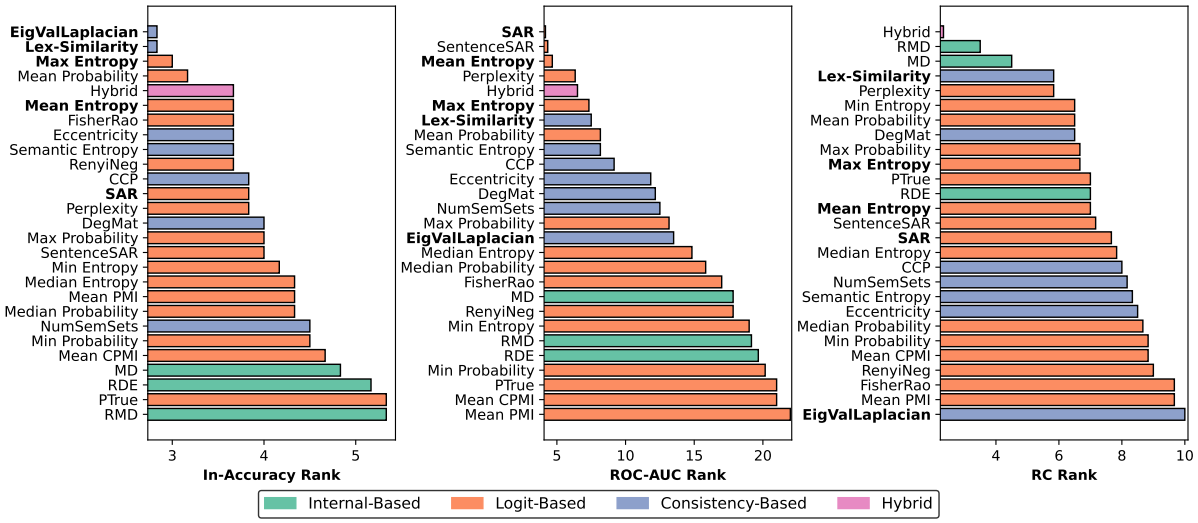

Figure 3: Uncertainty methods average ranks for In-Accuracy, ROC-AUC and Retrieval Calls. Smaller rank indicate average better performance. The In-Accuracy ranks demonstrate key downstream metrics, while the ROC-AUC ranks show self-knowledge abilities across different methods, affecting average downstream performance. The Retriever Calls (RC) ranks represent the efficiency of the method. This evaluation led to choose **EigValLaplacian**, **Lex-Similarity**, **Max Entropy**, **Mean Entropy**, and **SAR** for more detailed analysis.

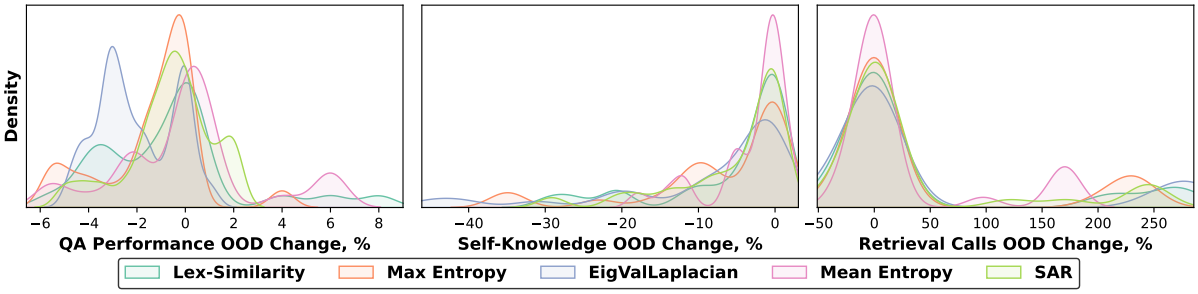

Figure 4: The transferability of methods between datasets was evaluated using average changes in metrics for Out-Of-Distribution (OOD) data. QA Performance in OOD was measured by InAccuracy, showing comparable results across methods. Self-Knowledge, evaluated by Accuracy, degraded significantly. Efficiency was assessed by RC, indicating that methods tend to call the retriever more frequently after transfer.

Appendix C), is used to assess OOD robustness. For statistical tests details, refer to Appendix A.

In Figure 4, we present the distributions of performance change across all train-test dataset pairs. For In-Accuracy, most methods perform comparably, with EigValLaplacian being the only method that significantly lags behind and differs from nearly all others. While most methods are centered around 0, indicating stability, there is a noticeable tail representing a loss in quality. Nevertheless, the loss typically remains under 4%, suggesting strong downstream OOD transfer performance, with occasional cases of positive improvement.

However, the QA performance can be influenced by multiple factors. Self-knowledge transfer, measured by Accuracy, reveals a complex picture. While the changes are centered around 0—an encouraging sign of stability—the tail indicating quality loss is notably heavier, with more extreme variations and no cases of positive transfer. EigValLaplacian stands out with the weakest transfer performance, whereas other methods show comparable results without statistically significant differences.

Efficiency transfer analysis shows a similar centering around 0 but reveals the largest percentage changes. Methods tend to call the retriever more frequently when transferred, indicating underconfidence. No significant differences are observed between methods.

Takeaway 4: UE methods show strong OOD robustness for QA performance but lower for self-knowledge and efficiency, with no significant differences between most methods.

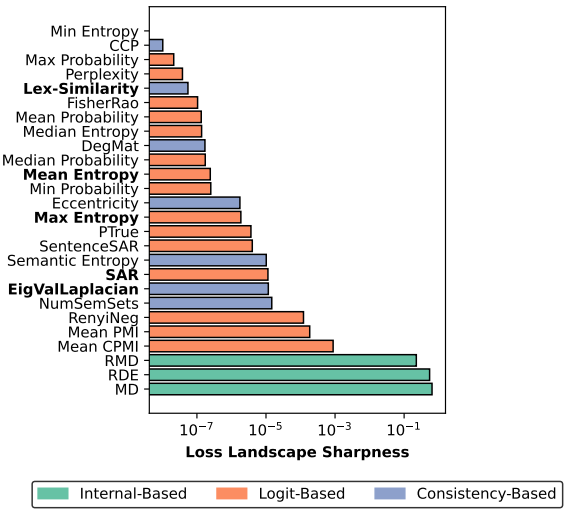

Figure 5: Average loss landscape sharpness in logarithmic scale. Higher values correspond to more complex functions.

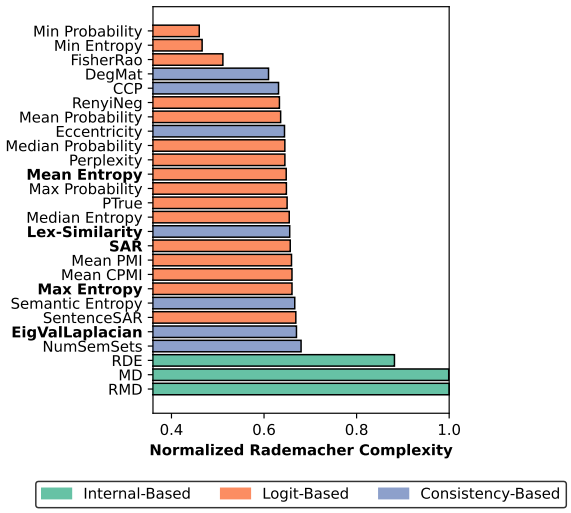

Figure 6: Normalized Rademacher Complexity for uncertainty methods. Higher values indicate richer complexity of feature.

7 Uncertainty Estimation Complexity

While many uncertainty estimation methods perform well in identifying self-knowledge and improving retrieval efficiency, their behavior often diverges in more challenging settings, such as multi-hop reasoning or out-of-domain transfer. One possible explanation lies in the complexity of the functions they induce. Complex uncertainty functions may overfit, generalize poorly, or lead to overconfident decisions.

To investigate this hypothesis, we assess the functional complexity of each uncertainty estimation method f by analyzing the behavior of a downstream classifier $C(f)$, trained to predict self-knowledge from the uncertainty scores. We employ 3 complementary complexity measures: Rademacher Complexity (Yin et al., 2019),Loss Landscape Sharpness (Sagun et al., 2016; Glorot and Bengio, 2010) and classifier sensitivity.

Rademacher Complexity quantifies the capacity of a hypothesis class to fit random noise with higher values indicating greater complexity:

$$\mathcal{R}_n(\mathcal{H}_f) = \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}_f} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right],$$

where $\mathcal{H}_f$ is the hypothesis class induced by uncertainty method f, $\sigma_i \sim \mathbb{U}\{-1, 1\}$ are Rademacher random variables, and $h(x_i)$ is the model's prediction.

Loss Landscape Sharpness quantifies complexity from different perspective evaluating the curvature of loss landscape, with higher values indicating more complex and harder generalizable functions (Kaur et al., 2023).

Let $\mathcal{L}(w) \in C^2$ be a twice continuously differentiable loss function with respect to $w \in \mathbb{R}^d$, and let $H(w) = \nabla_w^2 \mathcal{L}(w)$ denote its Hessian. The sharpness at the optimized parameters w^* is defined as:

$$\lambda_{\max} = \sup_{\|v\|_2 = 1} v^\top H(w^*) v,$$

where $\lambda_{\max}$ is the largest eigenvalue of $H(w^*)$, capturing the steepest curvature of the loss surface.

Classifier Sensitivity quantifies how sensitive is feature to the choice of classifier. We compute the average performance drop for each uncertainty method when switching from the maximum classifier performance to the average. This sensitivity further indicates the complexity of the method, as more complex methods require thorough choice of classifier and hyperparameters.

Resuts. Our analysis (Figures 5 6 and 7) reveals that internal-based uncertainty features exhibit the highest complexity, both in terms of functional expressivity and sharpness. Despite their conceptual appeal, these methods often struggle to generalize across datasets or complex questions. In contrast, logit-based and consistency-based methods show lower complexity and more stable performance, particularly in low-data or transfer scenarios.

We also observe that the Hybrid method exhibits the highest functional complexity. Due to its extreme values, it was excluded from Rademacher

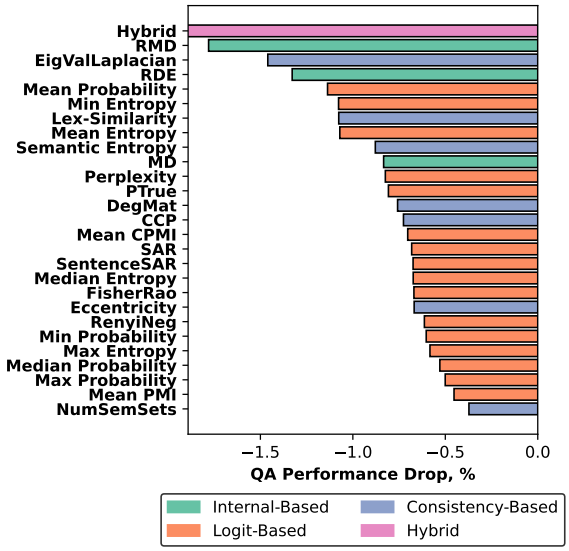

Figure 7: Average QA performance drop for uncertainty methods for when switching maximum over classifiers to average.

Complexity and Loss Landscape Sharpness. This outcome is expected: by combining all uncertainty signals, it introduces a highly expressive and over-parameterized feature space, which increases the overall complexity of the downstream classifier. Correspondingly, it also shows the largest drop in sensitivity, highlighting its potential overfitting and reduced robustness.

Takeaway 5: Hybrids and Internal-based methods are the most complex and harder to generalize, while consistency-based methods are more complex than logit-based ones.

8 Adaptive Retrieval for Usefulness

We additionally evaluate the adaptive retrieval efficiency and QA performance using a real-world proprietary setting aimed at enhancing usefulness for users. The evaluation dataset was collected by MTS AI, a commercial company developing AI-based solutions. It includes factual questions, contexts retrieved from the web, answers generated both with and without contextual information, usefulness annotations for these answers, and gold-standard answers composed by human experts.

Usefulness is a product-centric metric scored at discrete values of 0, 0.5, and 1, representing the degree of user-perceived value provided by an answer. The evaluation dataset consists of 1659 queries representing real-world questions posed by users to the system during a specific test period. These queries span various topics, all characterized by their factual nature, each seeking information on a specific granular fact.

Method	Usefulness	RC	Acc	AUC
Never RAG	0.442	0	-	-
Always RAG	0.736	1	-	-
EigValLaplacian	**0.735**	0.99	0.35	0.49
Lex-Similarity	0.443	0.01	0.33	0.52
Max Entropy	0.488	0.11	0.34	0.56
Mean Entropy	0.601	0.42	0.35	0.65
SAR	0.728	0.91	0.44	0.54
Hybrid	0.732	0.93	0.33	0.53
Ideal	0.743	0.48	-	-

Table 3: Comparison of UE methods tested on a usefulness task with proprietary dataset.

The evaluation results are presented in Table 3. The table demonstrates that UE methods can reduce retrieval calls without substantially compromising answer quality. Furthermore, density-based and hybrid methods show near-ideal usefulness can be achieved with a minor reduction in retrieval calls. These findings underscore the practical applicability of uncertainty estimation methods for adaptive retrieval in real-world scenarios.

9 Conclusion

We present a comprehensive computational study of adaptive retrieval systems, evaluating 27 established uncertainty estimation methods alongside 8 recently published methods tailored for this task. Our analysis considers downstream QA performance, efficiency, and self-knowledge, covering a total of 10 evaluation metrics. Our findings show that established uncertainty methods achieve performance comparable to recently proposed adaptive retrieval approaches, while being more efficient and exhibiting stronger self-knowledge capabilities.

Moreover, we conducted an in-depth comparison of the 27 uncertainty estimation methods, revealing notable discrepancies between downstream performance and self-knowledge metrics. Our analysis of OOD transfer shows minimal deviations in downstream performance but a significant decline in self-knowledge, with no substantial differences between methods. We also identify the higher functional complexity of internal-based methods.

Limitations

- We conduct our study using the LLaMA3.1-8b-instruct model, which is among the best open-source models within its parameter range. However, extending the analysis to additional models would help validate the consistency of our findings across different architectures.
- Our evaluation is performed on 6 QA datasets, which are standard for this task. Expanding the evaluation to include more QA datasets, particularly domain-specific ones, could uncover additional insights and highlight the generalizability of the methods.

Ethical Considerations

Text information retrieval systems may yield biased text documents, biasing the resulting generation of even an aligned ethically LLM in an undesired direction. Therefore, engineers deploying RAG and Adaptive RAG pipelines in real world applications facing users shall consider this potential issue.

Acknowledgments

This work was supported by the Russian Scientific Foundation project № 25-71-30008.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.1, knowledge storage and extraction. *Preprint*, arXiv:2309.14316.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jinheon Baek, Soyeong Jeong, Minki Kang, Jong C. Park, and Sung Ju Hwang. 2023. Knowledge-augmented language model verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1720–1736. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Maxime Darrin, Pablo Piantanida, and Pierre Colombo. 2023. Rainproof: An umbrella to shield text generator from out-of-distribution data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5831–5857. Association for Computational Linguistics.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *CoRR*, abs/2402.10612.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024a. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 9367–9385. Association for Computational Linguistics.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, et al. 2024b. Fact-checking the output of large language models via token-level uncertainty quantification. *arXiv preprint arXiv:2403.04696*.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. Lm-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*, pages 446–461. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7036–7050. Association for Computational Linguistics.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, Qun Liu, and Wei Wang. 2024. Learning to edit: Aligning llms with knowledge editing. *Preprint*, arXiv:2402.11905.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Simran Kaur, Jeremy Cohen, and Zachary Chase Lipton. 2023. On the maximum hessian eigenvalue and generalization. In *Proceedings on "I Can't Believe It's Not Better! - Understanding Deep Learning Through Empirical Falsification" at NeurIPS 2022 Workshops*, volume 187 of *Proceedings of Machine Learning Research*, pages 51–65. PMLR.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177.

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. 2020a.

Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick S. H. 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. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation. *arXiv preprint arXiv:2402.11457*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J. Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Levent Sagun, Leon Bottou, and Yann LeCun. 2016. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. 2024. Constructing benchmarks and interventions for combating hallucinations in llms. *Preprint*, arXiv:2404.09971.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024a. Dragin: Dynamic retrieval augmented generation based on the information needs of large language models. *Preprint*, arXiv:2403.10081.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024b. DRAGIN: dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12991–13013. Association for Computational Linguistics.

Junya Takayama and Yuki Arase. 2019. Relevant and informative response generation using pointwise mutual information. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 133–138.

Shuchang Tao, Liuyi Yao, Hanxing Ding, Yuexiang Xie, Qi Cao, Fei Sun, Jinyang Gao, Huawei Shen, and Bolin Ding. 2024. When to trust llms: Aligning confidence with response quality. *arXiv preprint arXiv:2404.17287*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5956–5965. Association for Computational Linguistics.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. In *Proceedings of the ACM Web Conference 2024*, pages 1362–1373.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2024. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *CoRR*, abs/2406.19215.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. 2019. Rademacher complexity for adversarially robust generalization. In *International conference on machine learning*, pages 7085–7094. PMLR.

Xunjian Yin, Xu Zhang, Jie Ruan, and Xiaojun Wan. 2024. Benchmarking knowledge boundary for large language models: A different perspective on model evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2270–2286. Association for Computational Linguistics.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3656–3672. Association for Computational Linguistics.

Method	Mean	Max	Difference
Hybrid	25.67	11.33	-14.33
RMD	23.50	15.00	-8.50
Perplexity	15.00	8.83	-6.17
MD	19.50	13.50	-6.00
CCP	15.50	9.67	-5.83
EigValLaplacian	11.67	6.00	-5.67
Median Entropy	15.67	10.67	-5.00
Mean Entropy	12.83	8.33	-4.50
RDE	16.83	12.67	-4.17
DegMat	13.67	10.00	-3.67
Max Entropy	10.00	6.50	-3.50
Mean CPMI	14.00	11.00	-3.00
Lex-Similarity	9.33	6.50	-2.83
SentenceSAR	12.00	9.17	-2.83
Min Entropy	11.83	9.33	-2.50
SAR	10.50	8.83	-1.67
PTrue	16.00	14.50	-1.50
Max Probability	11.33	10.50	-0.83
Min Probability	11.00	10.17	-0.83
FisherRao	9.17	8.50	-0.67
Semantic Entropy	9.17	8.50	-0.67
RenyiNeg	9.00	9.33	0.33
Mean Probability	5.83	6.33	0.50
Mean PMI	10.33	11.17	0.83
Median Probability	8.83	9.83	1.00
Eccentricity	7.83	8.83	1.00
NumSemSets	10.33	12.00	1.67

Table 4: Rank of UC methods by In-Accuracy, aggregated using the mean or maximum across different classifiers. A lower difference indicates reduced stability to classifier choice, whereas a higher difference reflects greater robustness to classifier choice.

A Statistical Tests for OOD Testing

To evaluate the OOD performance differences between uncertainty estimation methods under dataset-dependent conditions, we use the Friedman test, suitable for data with small sample sizes and no assumptions about normality, while also being appropriate for repeated measurements.

After the Friedman test, we apply the Nemenyi post-hoc test to identify statistically significant pairwise differences between methods, similarly due to rank-based nature and accounting for multiple comparisons to ensure robust analysis. We also report significance with asterisk atop of the number.

Method	NQ		SQUAD		TQA		2Wiki		HotPot		Musique	
	Over	Under	Over	Under	Over	Under	Over	Under	Over	Under	Over	Under
AdaptiveRAG	0.01	0.43	0.14	0.13	0.19	0.30	0.12	0.15	0.11	0.18	0.02	0.10
DRAGIN	0.00	0.45	0.00	0.18	0.00	0.64	0.00	0.32	0.00	0.29	0.00	0.11
FLARE	0.20	0.21	0.40	0.06	0.18	0.24	0.43	0.06	0.53	0.05	0.34	0.06
Rowen_{CL}	0.55	0.00	0.82	0.00	0.36	0.00	0.68	0.00	0.71	0.00	0.89	0.00
Rowen_{CM}	0.55	0.00	0.82	0.00	0.36	0.00	0.68	0.00	0.71	0.00	0.89	0.00
$\text{Rowen}_{\text{Hybrid}}$	0.55	0.00	0.82	0.00	0.36	0.01	0.68	0.00	0.71	0.00	0.89	0.00
Seakr	0.00	0.45	0.00	0.18	0.00	0.64	0.00	0.32	0.00	0.29	0.00	0.11
Lex-Similarity	0.01	0.21	0.00	0.15	0.18	0.06	0.00	0.28	0.02	0.22	0.00	0.10
Max Entropy	0.08	0.20	0.00	0.13	0.17	0.07	0.00	0.29	0.00	0.23	0.00	0.10
EigValLaplacian	0.03	0.33	0.00	0.17	0.21	0.08	0.00	0.30	0.01	0.23	0.00	0.10
SAR	0.06	0.29	0.00	0.16	0.18	0.09	0.00	0.28	0.03	0.22	0.00	0.11
Mean Entropy	0.04	0.29	0.00	0.14	0.14	0.07	0.00	0.29	0.00	0.15	0.00	0.10

Table 5: Over- and UnderConfidence for adaptive retrieval methods and uncertainty estimation. Values closest to zero indicate the best performance. UnderConfidence refers to cases where the method failed to detect self-knowledge despite its presence, while OverConfidence reflects cases where the method incorrectly detected self-knowledge when it was absent.

Method	NQ				SQUAD				TQA			
	EM	F1	InAcc	RC	EM	F1	InAcc	RC	EM	F1	InAcc	RC
CCP	0.398	0.512	0.496	0.94	0.252	0.389	0.312	1.00	**0.600**	**0.692**	**0.662**	0.28
DegMat	0.394	0.514	0.496	0.97	0.252	0.389	0.312	1.00	0.598	0.684	0.644	0.29
Eccentricity	0.404	0.520	0.500	0.84	0.252	0.390	0.312	1.00	0.594	0.677	0.638	0.21
EigValLaplacian	0.406	0.532	**0.512**	0.81	0.254	0.391	0.314	1.00	0.594	0.682	0.640	0.26
FisherRao	0.390	0.506	0.498	0.88	0.252	0.389	0.312	1.00	0.598	0.688	0.654	0.11
Hybrid	0.410	0.534	0.504	0.65	0.254	0.393	0.314	0.99	0.594	0.689	0.654	0.32
Lex-Similarity	0.420	**0.535**	**0.512**	0.58	0.256	**0.394**	**0.318**	0.96	**0.600**	0.689	0.646	0.22
MD	0.398	0.511	0.496	1.00	0.252	0.389	0.312	1.00	0.598	0.681	0.642	0.05
Max Entropy	**0.422**	**0.535**	0.506	0.73	0.252	0.389	0.312	1.00	0.598	0.689	0.650	0.22
Max Probability	0.418	0.532	0.502	0.82	0.252	0.389	0.312	1.00	0.592	0.683	0.646	0.21
Mean CPMI	0.390	0.506	0.496	1.00	0.252	0.389	0.312	1.00	0.592	0.675	0.640	0.02
Mean Entropy	0.402	0.514	0.498	0.88	0.254	0.392	0.314	0.95	0.598	0.687	0.650	0.30
Mean PMI	0.390	0.506	0.496	1.00	0.254	0.389	0.312	1.00	0.596	0.683	0.640	0.02
Mean Probability	0.404	0.512	0.498	0.77	0.258	**0.394**	**0.318**	0.98	0.592	0.681	0.642	0.06
Median Entropy	0.412	0.519	0.496	1.00	0.252	0.389	0.312	1.00	0.596	0.682	0.644	0.15
Median Probability	0.408	0.512	0.496	1.00	0.252	0.389	0.312	1.00	0.592	0.680	0.644	0.26
Min Entropy	0.398	0.515	0.504	0.93	0.252	0.389	0.312	1.00	0.592	0.675	0.636	0.00
Min Probability	0.398	0.515	0.502	0.91	0.252	0.389	0.312	1.00	0.592	0.675	0.636	0.00
NumSemSets	0.406	0.521	0.502	0.83	0.252	0.389	0.312	1.00	0.590	0.680	0.638	0.28
PTrue	0.388	0.506	0.496	1.00	0.252	0.389	0.312	1.00	0.592	0.676	0.636	0.00
Perplexity	0.404	0.515	0.498	0.77	0.256	0.392	0.316	0.98	0.594	0.683	0.646	0.16
RDE	0.388	0.506	0.496	1.00	0.252	0.389	0.312	1.00	0.588	0.670	0.634	0.08
RMD	0.394	0.508	0.496	1.00	0.252	0.389	0.312	1.00	0.592	0.675	0.636	0.00
RenyiNeg	0.402	0.517	0.498	0.96	0.252	0.389	0.312	1.00	0.594	0.688	0.654	0.24
SAR	0.410	0.526	0.500	0.79	0.254	0.389	0.312	1.00	0.590	0.681	0.642	0.29
Semantic Entropy	0.406	0.521	0.504	0.83	**0.260**	0.393	0.316	0.92	0.596	0.685	0.640	0.24
SentenceSAR	0.410	0.521	0.500	0.73	0.254	0.391	0.314	0.99	0.596	0.685	0.644	0.24

Table 6: Detailed QA performance results for uncertainty methods on one-hop datasets. 'InAcc' denotes In-Accuracy, and 'EM' stands for Exact Match. Higher values indicate better performance. Bold values highlight the best results. Standard deviations for InAcc, EM, and F1 are $\approx 0.02 \pm 0.003$, calculated using bootstrapping.

B Performance Analysis Across Datasets

The scatter plot visualizes the performance comparison of various Retrieval-Augmented Generation (RAG) methods for all studied datasets, Figure 8. The x-axis represents the number of LLM calls, while the y-axis shows the Bootstrap Mean In-Accuracy. Circle sizes in the visualization correspond to the number of retrieval calls required by each method.

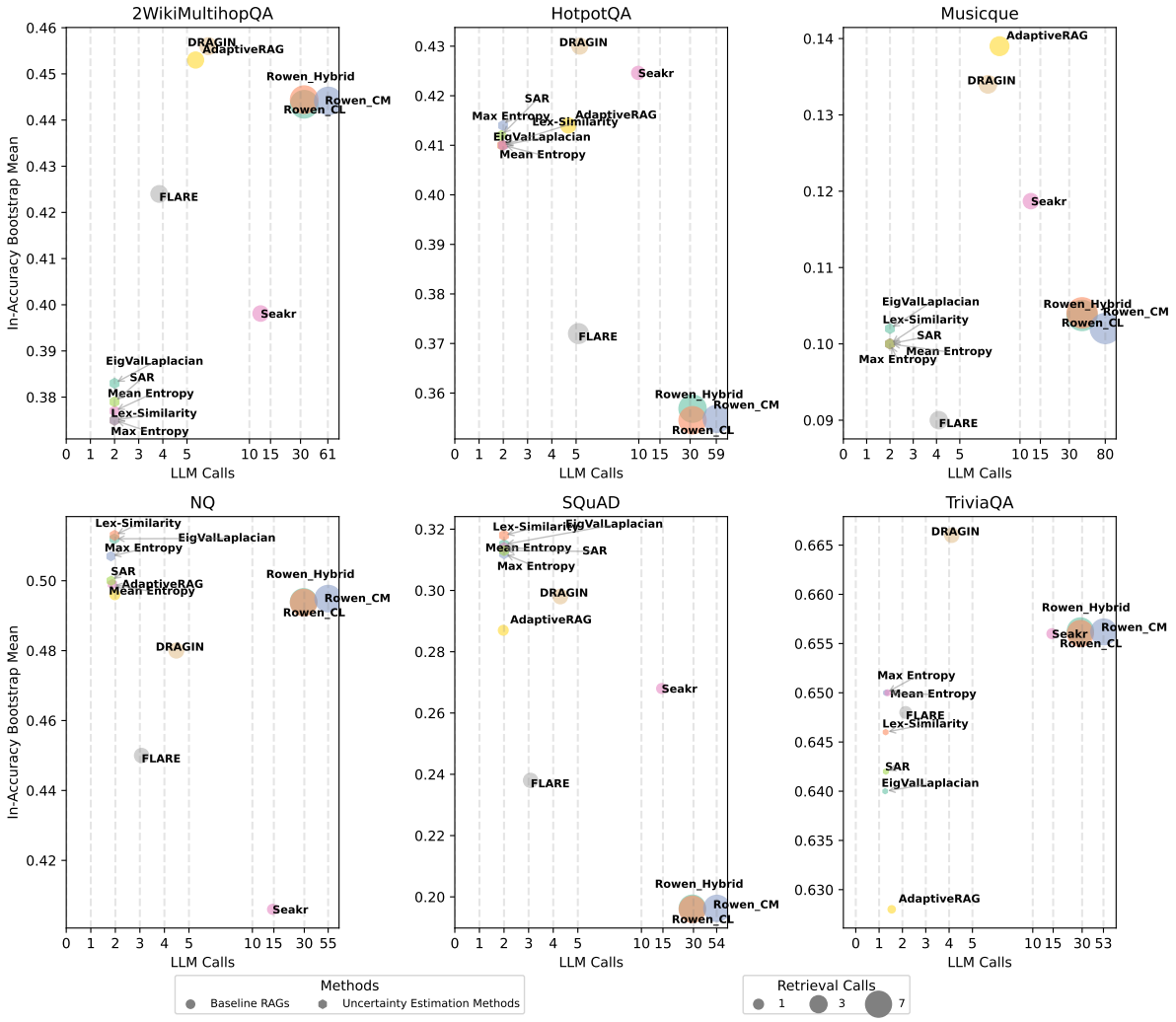

Figure 8: Performance comparison showing the relationship between LLM calls and Bootstrap Mean In-Accuracy. The size of each point indicates the number of retrieval calls required by each method.

Method	2Wiki				HotPot				Musique			
	EM	F1	InAcc	RC	EM	F1	InAcc	RC	EM	F1	InAcc	RC
CCP	0.310	0.398	0.376	0.98	0.386	0.497	0.410	1.00	0.088	0.167	0.100	1.00
DegMat	**0.314**	**0.407**	0.382	0.95	0.386	0.498	0.410	1.00	0.088	0.168	0.100	1.00
Eccentricity	0.312	0.406	**0.384**	0.93	**0.390**	**0.502**	**0.414**	0.93	0.088	0.167	0.100	1.00
EigValLaplacian	0.312	0.405	**0.384**	0.98	0.384	0.501	0.410	0.91	0.088	0.169	0.102	1.00
FisherRao	0.306	0.399	0.378	0.98	0.386	0.497	0.410	1.00	0.088	0.169	0.100	1.00
Hybrid	0.298	0.391	0.368	0.93	0.384	0.491	0.406	0.94	0.090	0.169	0.102	1.00
Lex-Similarity	0.306	0.400	0.376	0.97	0.386	0.498	0.410	0.95	0.088	0.168	0.100	1.00
MD	0.302	0.397	0.374	1.00	0.386	0.497	0.410	1.00	0.088	0.167	0.100	1.00
Max Entropy	0.304	0.398	0.376	0.95	**0.390**	0.501	**0.414**	0.99	0.088	0.167	0.100	1.00
Max Probability	0.304	0.396	0.374	1.00	0.386	0.497	0.410	1.00	0.088	0.167	0.100	1.00
Mean CPMI	0.302	0.397	0.376	0.98	0.386	0.497	0.410	1.00	0.090	0.169	0.102	0.99
Mean Entropy	0.306	0.400	0.378	0.93	0.386	0.497	0.410	0.99	0.088	0.167	0.100	1.00
Mean PMI	0.310	0.399	0.382	0.96	0.386	0.497	0.410	1.00	0.088	0.167	0.100	1.00
Mean Probability	0.308	0.400	0.380	0.96	0.388	0.498	0.412	0.97	**0.092**	**0.173**	**0.104**	0.97
Median Entropy	0.308	0.398	0.378	0.98	0.386	0.497	0.410	1.00	0.088	0.167	0.100	1.00
Median Probability	0.304	0.397	0.376	0.94	0.386	0.497	0.410	1.00	0.090	0.169	0.102	1.00
Min Entropy	0.308	0.397	0.376	0.93	0.386	0.497	0.410	1.00	0.090	0.171	**0.104**	0.99
Min Probability	0.312	0.401	0.376	0.95	0.386	0.497	0.410	1.00	0.090	0.169	0.102	0.99
NumSemSets	0.304	0.396	0.374	1.00	0.386	**0.502**	0.412	0.95	0.088	0.167	0.100	1.00
PTrue	0.308	0.398	0.372	0.87	0.386	0.497	0.410	1.00	0.090	0.169	0.102	0.99
Perplexity	0.304	0.398	0.376	0.96	0.386	0.498	0.410	1.00	0.088	0.168	0.100	1.00
RDE	0.304	0.398	0.376	0.99	0.386	0.497	0.410	1.00	0.090	0.171	0.102	0.99
RMD	0.304	0.398	0.372	0.97	0.388	0.499	0.412	0.95	0.088	0.167	0.100	1.00
RenyiNeg	0.302	0.396	0.374	1.00	**0.390**	0.500	**0.414**	0.97	0.088	0.167	0.100	1.00
SAR	0.310	0.404	0.380	0.97	0.386	0.500	0.412	0.90	0.088	0.167	0.100	1.00
Semantic Entropy	0.304	0.398	0.374	1.00	0.386	0.499	0.412	0.93	0.088	0.169	0.102	1.00
SentenceSAR	0.308	0.403	0.376	0.89	0.384	0.498	0.410	0.90	0.088	0.167	0.100	1.00

Table 7: Detailed QA performance results for uncertainty methods on one-hop datasets. 'InAcc' denotes In-Accuracy, and 'EM' stands for Exact Match. Higher values indicate better performance. Bold values highlight the best results. Standard deviations for InAcc, EM, and F1 are $\approx 0.02 \pm 0.002$ for HotPotQA and 2Wiki and $\approx 0.01 \pm 0.001$ for Musique, calculated using bootstrapping.

Method	NQ					SQUAD					TQA				
	EM	F1	Acc	LLMC	RC	EM	F1	Acc	LLMC	RC	EM	F1	Acc	LLMC	RC
No Context	0.386	0.495	0.446	1.0	0.00	0.156	0.249	0.176	1.0	0.00	0.592	0.675	0.636	1.0	0.00
All Context	0.388	0.506	0.496	1.0	1.00	0.252	0.389	0.312	1.0	1.00	0.522	0.636	0.610	1.0	1.00
AdaptiveRAG	0.388	0.505	0.496	1.0	0.98	0.238	0.366	0.286	1.0	0.97	0.564	0.656	0.628	0.5	0.54
DRAGIN	0.396	0.510	0.480	4.5	2.24	0.244	0.371	0.298	4.3	2.14	0.584	0.691	0.666	4.1	2.06
FLARE	0.358	0.477	0.450	3.1	2.07	0.190	0.303	0.238	3.1	2.08	0.570	0.674	0.648	2.1	1.39
FS-RAG	0.348	0.483	0.428	2.7	2.70	0.226	0.361	0.286	2.8	2.78	0.540	0.640	0.632	2.5	2.47
IRCoT	0.392	0.502	0.478	2.7	2.70	0.210	0.341	0.268	2.7	2.68	0.526	0.634	0.608	2.7	2.74
Rowen_{CL}	0.002	0.104	0.494	29.5	7.24	0.004	0.061	0.196	29.2	7.19	0.022	0.188	0.656	28.7	7.06
Rowen_{CM}	0.002	0.104	0.494	29.5	7.27	0.004	0.061	0.196	29.2	7.20	0.022	0.188	0.656	28.7	7.12
$\text{Rowen}_{\text{Hybrid}}$	0.002	0.104	0.494	55.0	7.27	0.004	0.061	0.196	54.3	7.15	0.022	0.189	0.656	53.4	6.93
Seakr	0.360	0.487	0.406	14.6	1.00	0.226	0.361	0.268	14.6	1.00	0.598	0.692	0.656	14.6	1.00

Table 8: Results of baselines for onehop datasets. LLMC refers to the average number of LLM calls per question, while RC indicates the average number of retrieval calls per question. For NQ the standard deviations of Acc, EM, and F1 are $\approx 0.022 \pm 0.001$ across all methods. For SQUAD and Trivia the standard deviations of Acc, EM, and F1 are $\approx 0.018 \pm 0.006$ across all methods. Overall, the methods exhibit similar deviations, with Rowen showing the lowest deviation, typically ≤ 0.01.

Method	2Wiki					HotPotQA					Musique				
	EM	F1	Acc	LLMC	RC	EM	F1	Acc	LLMC	RC	EM	F1	Acc	LLMC	RC
No Context	0.302	0.371	0.318	1.0	0.00	0.280	0.372	0.286	1.0	0.00	0.100	0.193	0.106	1.0	0.00
All Context	0.302	0.396	0.374	1.0	1.00	0.386	0.497	0.410	1.0	1.00	0.088	0.167	0.100	1.0	1.00
AdaptiveRAG	0.384	0.471	0.454	2.6	2.64	0.396	0.499	0.414	2.3	2.34	0.122	0.216	0.140	3.6	3.63
DRAGIN	0.406	0.480	0.456	5.8	2.92	0.398	0.506	0.430	5.1	2.56	0.116	0.207	0.134	6.3	3.15
FLARE	0.358	0.451	0.424	3.9	2.85	0.298	0.391	0.372	5.1	4.07	0.076	0.161	0.090	4.1	3.10
FS-RAG	0.348	0.431	0.388	3.8	3.76	0.376	0.503	0.422	3.7	3.70	0.088	0.187	0.100	3.4	3.35
IRCoT	0.362	0.460	0.454	4.4	4.38	0.414	0.516	0.438	3.5	3.45	0.116	0.221	0.138	4.1	4.08
Rowen_{CL}	0.002	0.083	0.444	32.9	7.87	0.002	0.084	0.354	31.9	7.67	0.002	0.034	0.104	42.1	9.52
Rowen_{CM}	0.002	0.083	0.444	32.9	7.87	0.002	0.084	0.356	31.9	7.70	0.002	0.034	0.104	42.1	9.52
$\text{Rowen}_{\text{Hybrid}}$	0.002	0.083	0.444	61.8	7.85	0.004	0.086	0.354	59.8	7.63	0.002	0.034	0.102	80.2	9.48
Seakr	0.382	0.460	0.398	12.3	2.44	0.400	0.523	0.424	9.9	1.76	0.112	0.215	0.118	12.3	2.40

Table 9: Results of baselines for multihop datasets. LLMC refers to the average number of LLM calls per question, while RC indicates the average number of retrieval calls per question. For 2Wiki and HotPotQA, the standard deviations of Acc, EM, and F1 are $\leq 0.022 \pm 0.001$ across all methods. For Musique, the standard deviations are $\leq 0.015 \pm 0.001$. Overall, the methods exhibit similar deviations, with Rowen showing the lowest deviation, typically ≤ 0.01.

	NQ	SQUAD	TQA	2Wiki	HotPot	Musique	Avg
Mean CPMI	-2.02	-7.44	-4.38	-2.98	-5.76	0.78	-3.63
Mean PMI	-1.45	-8.21	-4.69	-4.19	-5.37	3.20	-3.45
RDE	-1.29	-7.18	-3.52	-2.23	-4.68	1.18	-2.95
PTrue	-1.94	-7.95	-3.77	-2.03	-5.07	3.53	-2.87
EigValLaplacian	-3.28	-7.01	-3.44	-2.29	-3.32	2.35	-2.83
Min Probability	-1.83	-5.26	-3.52	-2.98	-3.22	1.96	-2.48
RenyiNeg	-1.04	-4.23	-6.12	0.21	-3.38	2.40	-2.03
NumSemSets	-0.96	-5.90	-3.39	-0.21	-3.20	1.60	-2.01
FisherRao	-0.80	-5.26	-3.44	-2.75	-3.12	3.60	-1.96
Min Entropy	-2.22	-4.23	-3.08	-0.85	-1.17	0.38	-1.86
Median Entropy	-0.89	-3.97	-3.81	-1.48	-3.80	3.60	-1.72
Median Probability	-1.13	-2.44	-4.60	-0.11	-3.41	1.96	-1.62
Hybrid	-1.83	-4.71	-4.28	-0.32	-5.12	7.06	-1.53
Mean Probability	-0.16	-2.39	-4.22	-0.42	-2.43	1.54	-1.35
Max Entropy	-1.90	-2.56	-4.80	-0.43	-3.00	6.40	-1.05
CCP	0.32	-2.18	-6.77	-0.64	-2.44	6.00	-0.95
Max Probability	-0.64	-2.95	-4.41	-0.64	-2.24	5.20	-0.95
DegMat	0.56	-2.69	-3.91	-1.57	-2.15	4.80	-0.83
Lex-Similarity	-2.50	-3.77	-3.41	-0.85	-2.34	8.00	-0.81
Eccentricity	0.00	-2.31	-2.63	-2.60	-2.71	5.60	-0.78
SAR	-0.24	-2.31	-2.87	-1.05	-3.11	5.60	-0.66
SentenceSAR	0.32	-2.42	-3.35	-0.85	-2.15	5.20	-0.54
Semantic Entropy	-0.48	-1.77	-2.69	0.43	-1.94	3.53	-0.49
RMD	0.08	-7.56	-1.76	-0.11	-0.49	7.60	-0.37
Perplexity	0.24	-2.66	-4.27	0.85	-1.56	5.60	-0.30
Mean Entropy	-0.48	-1.40	-3.94	0.74	-1.76	7.20	0.06
MD	0.00	-2.56	-2.74	0.65	0.00	9.60	0.82

Table 10: Average QA performance differences after transfer (in percentage) for each dataset. Negative values indicate a loss in In-Accuracy compared to in-domain testing, while positive values represent an In-Accuracy gain.

Method acronym	Method full name	Short description
	logit based	
FisherRao (Darrin et al., 2023)	Fisher-Rao distance	FisherRao is a distance on the Riemannian space formed by the parametric distributions, using the Fisher information matrix as its metric. It computes the geodesic distance between two discrete distributions.
Max Entropy (Fomicheva et al., 2020)	Maximum Token Entropy	The maximum entropy of all tokens in the generated sequence.
Max Probability	Maximum Sequence Probability	The score leverages the probability of the most likely sequence generation.
Mean CPMI (van der Poel et al., 2022)	Mean conditional pointwise mutual information	Extension of the PMI method by considering only those marginal probabilities for which the entropy of the conditional distribution is above certain threshold.
Mean Entropy (Fomicheva et al., 2020)	Mean Token Entropy	The average entropy of each individual token in the generated sequence.
Mean PMI (Takayama and Arase, 2019)	Mean pointwise mutual information	PMI compares the probability of two events (the question and the generated answer) occurring together to what this probability would be if the events were independent.
Mean Probability	Mean Sequence Probability	The total uncertainty is measured via average sequence probability.
Median Entropy (Fomicheva et al., 2020)	Median Token Entropy	The median entropy of all tokens in the generated sequence.
Median Probability	Median Sequence Probability	The total uncertainty is measured via median sequence probability.
Min Entropy (Fomicheva et al., 2020)	Minimum Token Entropy	The minimum entropy of all tokens in the generated sequence.
Min Probability	Minimum Sequence Probability	The score leverages the probability of the least likely sequence generation.
Perplexity (Fomicheva et al., 2020)	Perplexity	The score computes the average negative log probability of generated tokens, which is further exponentiated.
PTrue (Kadavath et al., 2022)	probability P(true)	The method measures the uncertainty of the claim by asking the LLM itself whether the generated claim is true or not. The confidence is the probability of thefirst generated token y1 being equal to "True".
RenyiNeg (Darrin et al., 2023)	Rényi negentropy	The score computes alpha-Renyi-divergence between the sample and the uniform distributions.
SAR (Duan et al., 2023)	Shifting Attention to more Relevant	SAR corrects generative inequalities by reviewing the relevance of each token and emphasizing uncertainty quantification attention to those more relevant components. The relevance is measured by calculating similarity of sentence before and after removing the certain token.
SentenceSAR (Duan et al., 2023)	Shifting Attention to more Relevant at Sentence level	SAR measured at sentence-level.
	consistency based	
CCP (Fadeeva et al., 2024a)	Claim-Conditioned Probability	The method aggregates token-level uncertainties into a claim-level score, it removes the impact of uncertainty about what claim to generate on the current step and what surface form to use.
DegMat (Lin et al., 2023)	Degree matrix	Using the Degree matrix a new uncertainty measure could be found that reflects the average pairwise distance.
Eccentricity (Lin et al., 2023)	Eccentricity	The smallest k eigenvectors of Laplacian Graph are used as the proxy for the models' embeddings. Then, we could use the average offset from the average embedding as the uncertainty measure.
EigValLaplacian (Lin et al., 2023)	Sum of eigenvalues of the graph Laplacian	The score uses pairwise similarities between the sampled answers to the questions to form the symmetric weighted adjacency matrix (degree matrix). This matrix is further used to create the graph Laplacian. The sum of Eigenvalues of the Graph Laplacian are used as a measure of uncertainty.
Lex-Similarity (Fomicheva et al., 2020)	Lexical similarity	The score computes how similar two words or phrases are in terms of their meaning.
NumSemSets (Lin et al., 2023)	Number of semantic sets	The number of semantic sets initially equals the total number of generated answers K. If two answers are semantically similar, they are put into one cluster. A higher number of semantic sets corresponds to an increased level of uncertainty, as it suggests a higher number of diverse semantic interpretations for the answer.
Semantic Entropy (Kuhn et al., 2023)	Semantic Entropy	The method aims to deal with the generated sequences that have similar meaning while having different probabilities according to the model. The idea is to cluster generated sequences into several semantically homogeneous clusters with a bi-directional entailment algorithm and average the sequence probabilities within the clusters.
	internal-based	
MD (Lee et al., 2018)	Mahalanobis distance	In this paper, the authors propose a simple yet effective method for detecting any abnormal samples, which is applicable to any pre-trained softmax neural classifier. They obtain the class conditional Gaussian distributions with respect to (low- and upper-level) features of the deep modelsunder Gaussian discriminant analysis, which result in a confidence score based on the Mahalanobis distance.
RDE (Yoo et al., 2022)	Robust density estimation	The method improves over MD by reducing the dimensionality of the last hidden state of the decoder averaged over all generated tokens via PCA decomposition. Additionally, computing of the covariance matrix for each individual class is done by using the Minimum Covariance Determinant estimation. The uncertainty score is computed as the MD in the space of reduced dimensionality.
RMD (Ren et al., 2023)	Relative Mahalanobis distance	The MD distance score is adjusted by subtracting from it the other MD score computed for some large general purpose dataset covering many domains.
	blended approach	
Hybrid	Hybrid	Our hybrid approach that uses all uncertainty features defined in the table.

Table 11: Description of the uncertainty estimation methods used in the paper. The methods are grouped by their categories: logit based, consistency-based, internal-based and hybrid.

Method	EigValLaplacian	Lex-Similarity	Max Entropy	Mean Entropy	SAR
EigValLaplacian	1.00	0.03	0.81	0.00	0.00
Lex-Similarity	0.03	1.00	0.38	0.42	0.95
Max Entropy	0.81	0.38	1.00	0.00	0.08
Mean Entropy	0.00	0.42	0.00	1.00	0.86
SAR	0.00	0.95	0.08	0.86	1.00

Table 12: In-Accuracy P-Value, Friedman Test Results: Test Statistic: 29.580 P-value: 0.00001

Method	EigValLaplacian	Lex-Similarity	Max Entropy	Mean Entropy	SAR
EigValLaplacian	1.00	0.00	0.08	0.00	0.05
Lex-Similarity	0.00	1.00	0.81	0.94	0.88
Max Entropy	0.08	0.81	1.00	0.33	1.00
Mean Entropy	0.00	0.94	0.33	1.00	0.42
SAR	0.05	0.88	1.00	0.42	1.00

Table 13: Accuracy P-Value, Friedman Test Results: Test Statistic: 22.847 P-value: 0.00014

C Performance Analysis Across OOD Datasets

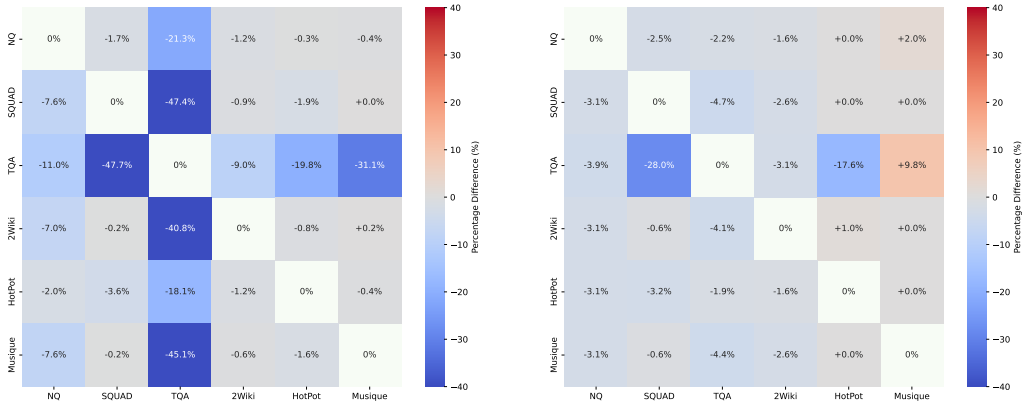

Figure 9: Heatmap of improvement/decrease of the Accuracy and In-Accuracy scores on the OOD setup for the EigValLaplacian method.

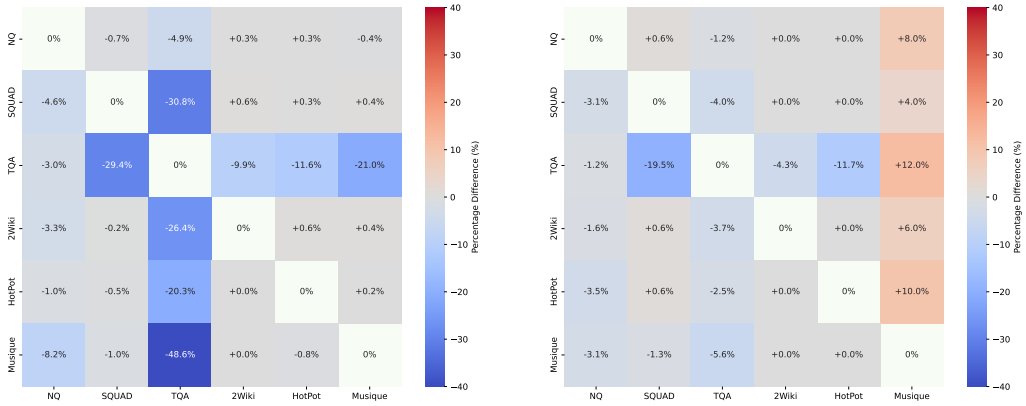

Figure 10: Heatmap of improvement/decrease of the Accuracy and In-Accuracy scores on the OOD setup for the Lex-Similarity method.

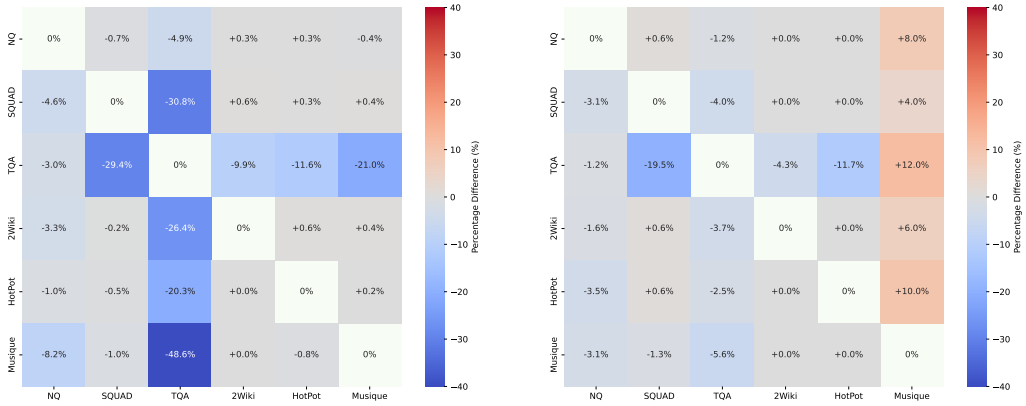

Figure 11: Heatmap of improvement/decrease of the Accuracy and In-Accuracy scores on the OOD setup for the MaxEntropy method.

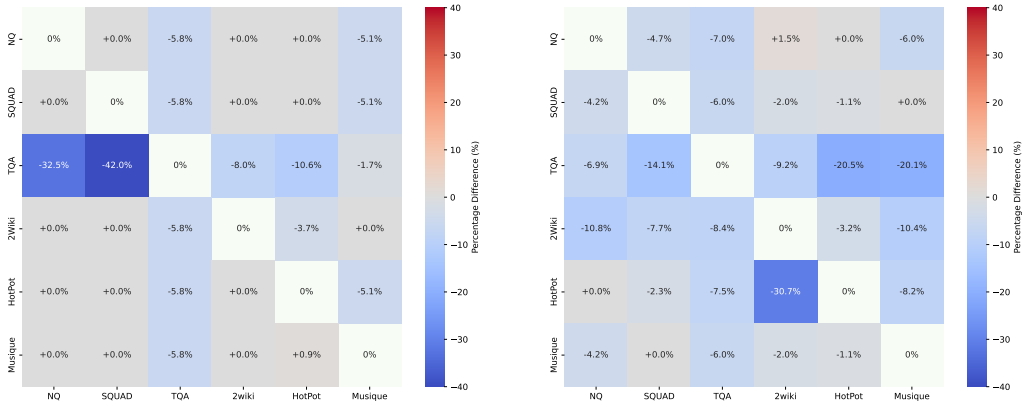

Figure 12: Heatmap of improvement/decrease of the In-Accuracy scores on the OOD setup for the SeaKR and DRAGIN methods

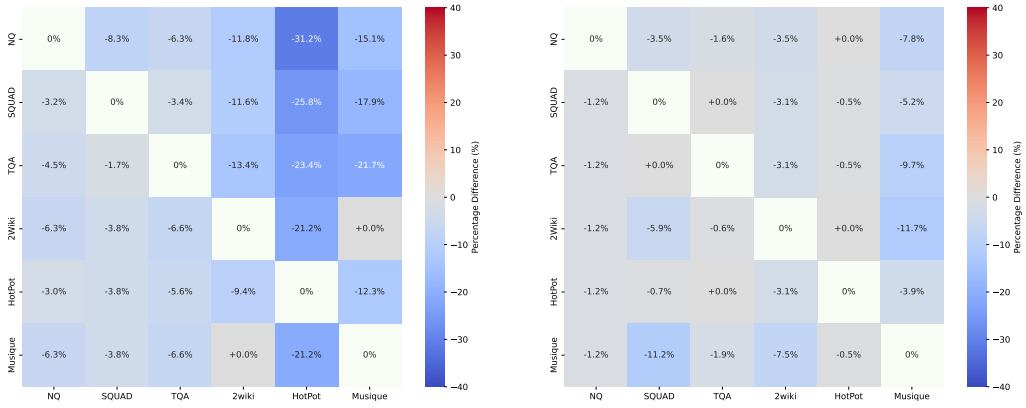

Figure 13: Heatmap of improvement/decrease of the In-Accuracy scores on the OOD setup for the FLARE and AdaptiveRAG methods. AdaptiveRAG shows the most stable performance in OOD.

D Feature Importance Analysis for Hybrid Method of Uncertainty Estimation

This section provides a figure 14 to represent ranks importance of different uncertainty estimation methods as a feature in a hybrid method. In addition, Figure 15 represents feature importance estimation in the form of a bar chart for each dataset.

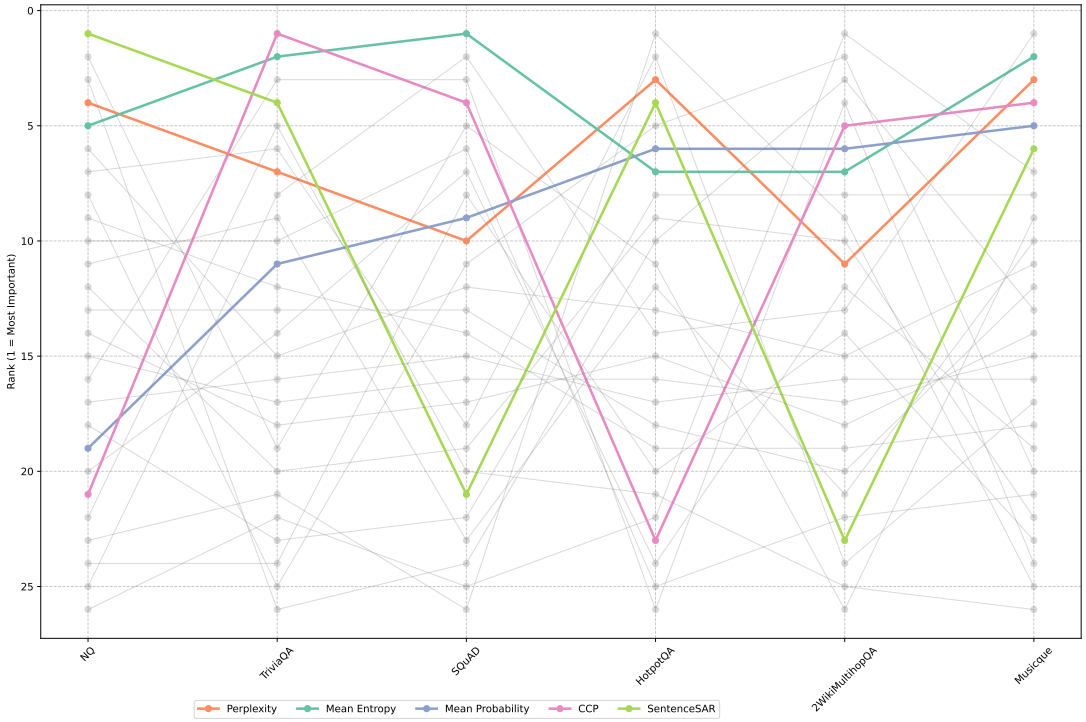

Figure 14: Top-5 UE methods as a features for hybrid method across datasets.

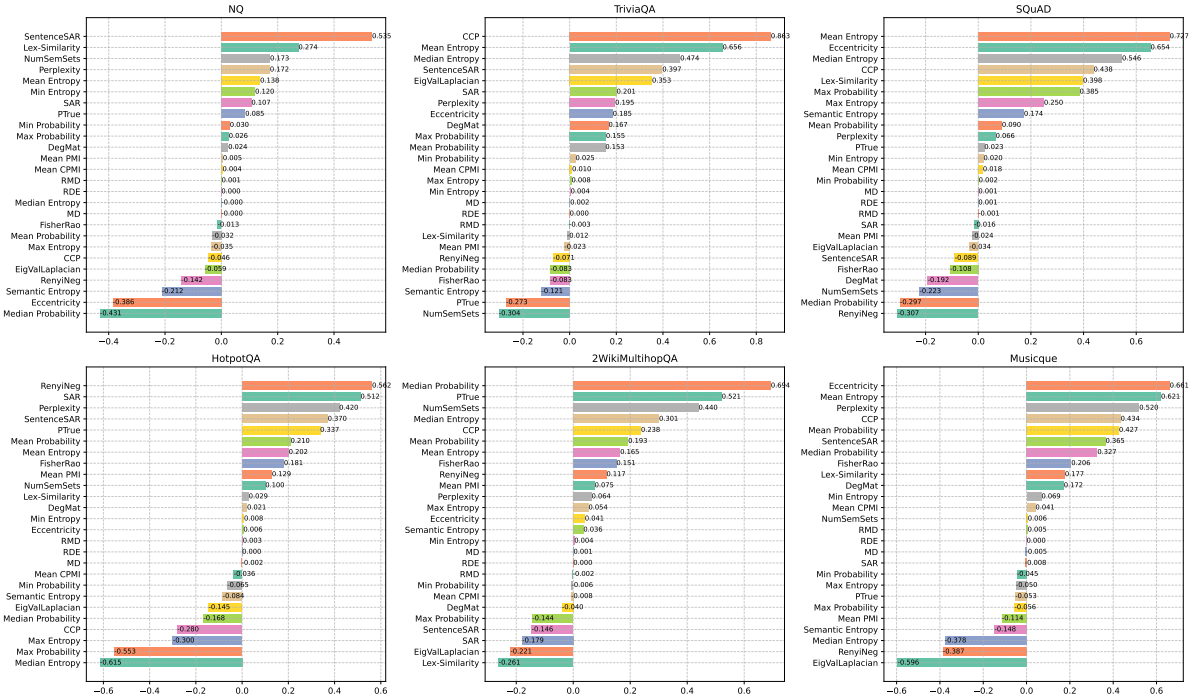

Figure 15: Feature Importance for each dataset for Hybrid method.

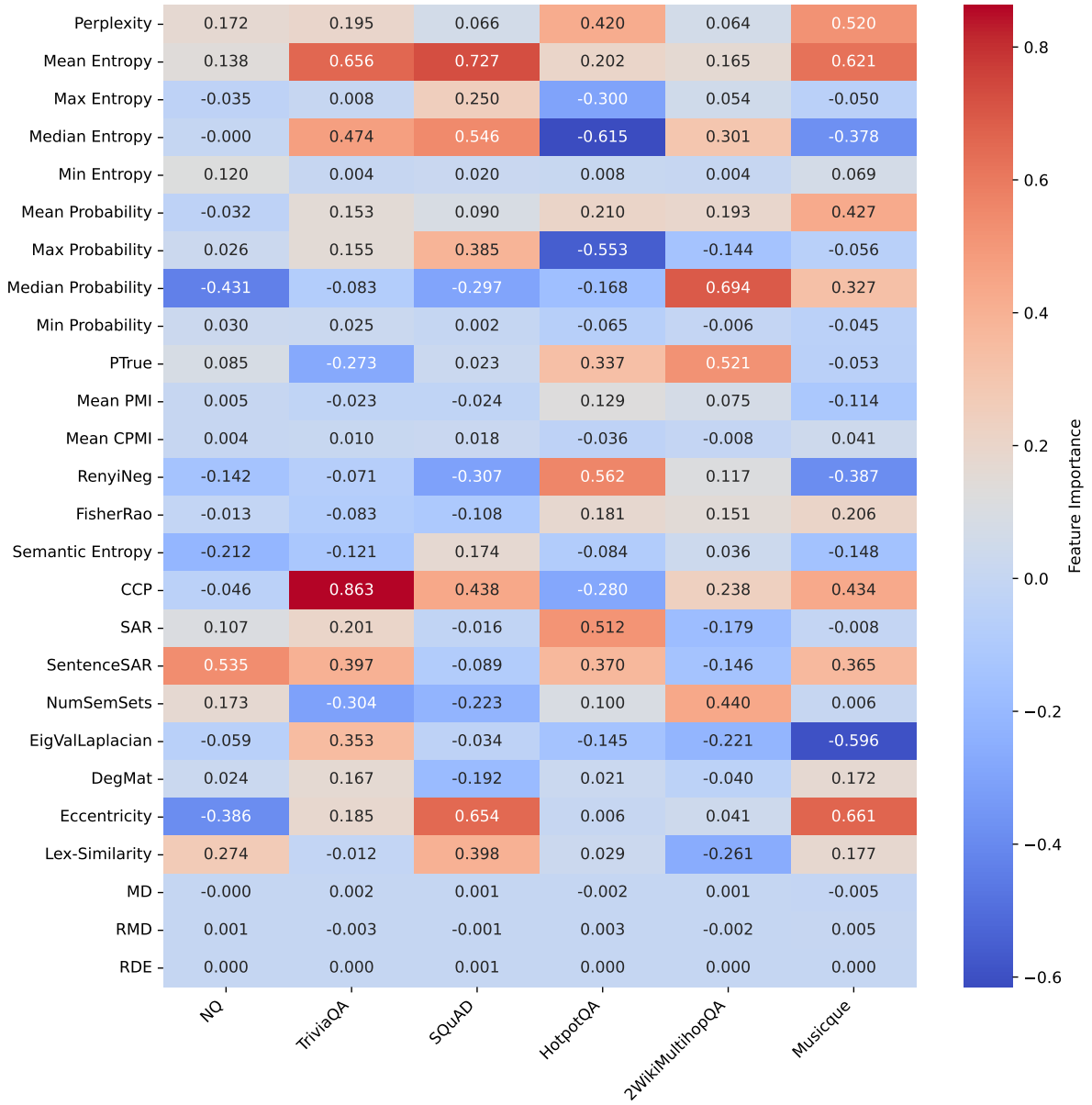

Figure 16: Feature Importance across datasets for Hybrid method. Different Uncertainty Estimation methods showed different performance on different datasets.

E Technical Details

For all experiments, we use the LLaMA 3.1-8b-instruct model with its default generation parameters. In our baseline methods, we strictly adhere to their original procedures, including prompting, parameter settings, and other configurations. For testing uncertainty estimation methods, we follow the protocol of AdaptiveRAG (Jeong et al., 2024), using the same prompt and few-shot examples. For the Rowen Consistency Model evaluation we use Qwen 2.5-72B-Instruct (Yang et al., 2024; Team, 2024) as the verification model instead of the original Qwen-Max-0428 due to the API usage limitations.

For all our methods we use the same retriever, a term-based sparse retrieval model known as BM25 (Robertson et al., 1994) and the same version of Elasticsearch 7.17.9[3], following previous studies (Su et al., 2024b; Yao et al., 2024). For the external document corpus, we use the Wikipedia corpus preprocessed by Karpukhin et al. (2020).

For all uncertainty methods, we compute scores on both the training and test sets with LM-Polygraph (Fadeeva et al., 2023) with MIT License. Using the training set scores, we fit multiple classifiers, including Threshold, Logistic Regression, Decision Tree, KNN, and MLP. The performance of the best classifier is reported based on downstream metrics, with a further analysis of classifier stability provided in Section **??**.

For classifiers, we employed scikit-learn library (Buitinck et al., 2013) and the following configurations:

- Logistic Regression with default hyperparameters.
- Threshold Classifier is optimized by finding the best threshold for In-Accuracy over a log-scaled grid of size 200, spanning the minimum to maximum training uncertainty values.
- Decision Tree with a maximum depth of 3.
- K-Nearest Neighbors (KNN) using 15 neighbors.
- Multi-Layer Perceptron (MLP) configured with 2 hidden layers, each of size 64.

All hyperparameters remained fixed across all runs to ensure consistency.

Standard deviation is calculated via bootstrap sampling using 1000 rounds.

For trainable uncertainty methods, such as Mahalanobis Distance, we split the training data into two equal parts: one part is used to learn the parameters of the uncertainty method, while the other is used to train the classifier. For Relative Mahalanobis Distance, we utilize C4 as the source of additional relative data for training the parameters.

Our evaluation was conducted on an NVIDIA A100 GPU. The total runtime was approximately 6 hours for SeaKR, 18 hours for both DRAGIN and FLARE, 36 hours for IRCoT, 2 hours for training AdaptiveRAG on IRCoT generations, and 10 hours for Rowen_{CM}, Rowen_{CL}, and Hybrid (with caching). In contrast, all uncertainty estimations at once required less than 1 hour, highlighting their computational efficiency and reduced CO_2 emissions.

F Methods

IRCoT (Trivedi et al., 2023) – **I**nterleaves **R**etrieval in a **CoT** was one of the pioneering methods to work with multi-hop questions. The authors proposed a new approach that interleaves retrieval with steps in chain-of-thought (CoT) reasoning. At first, the authors rertrieve K paragraphs relevant to the question Q as a query. Next, there are two steps, namely, `reason` and `retrieve` that are made iteratively until the termination criterion is met. As the incontext examples, questions, answers, gold relevant contexts and the example of CoT for the question are shown. In the `reason` step, we show the model CoT reasoning generated so far and let it complete the rest. Although the model can generate multiple sentences in the CoT, only the first generated sentence is taken. If the CoT contains the phrase "answer is:" (which was shown in the context examples as a phrase after which the final answer is written, so that we fix the format

[3] https://www.elastic.co/guide/en/elasticsearch/reference/7.17/release-notes-7.17.9.html

of the answers) or the maximum number of iterations has been reached [4], the process is terminated. In the `retrieve` step the last generated sentence in the CoT is taken to retrieve more additional paragraphs that would be relevant to answer the questions. These newly retrieved paragraphs are added to the ones retrieved in the previous question [5] as the context for the question.

Adaptive RAG (Jeong et al., 2024) uses the complexity of the question for adaptive retrieval. Simple questions can be answered without retrieval at all while complex questions require a multistep approach with iterative usage of both LLMs and retrievers. While users often ask simple and straightforward questions, the strategy which is necessary for answering complex questions is largely inefficient for the simple queries. The authors proposed a balanced strategy by training a classifier that predicts one of the three outcomes: whether not to retrieve at all (class A), retrieve once (class B) and retrieve multiple times (class C, the authors use IRCoT (Trivedi et al., 2023)). The classifier based on the t5-large model is trained on the development parts of the six considered datasets. The authors ran questions for all three methods and labeled as the correct the most efficient one. The most efficient means that if the correct answer is obtained by all three classes, class A is returned as the true one. As the additional training data the authors used the inductive biases in datasets (this concept assumes that simple questions should be answered with one step retrieve, and complex questions with multistep retrieve).

FLARE (Jiang et al., 2023) – **F**orward-**L**ooking **A**ctive **Re**trieval augmented generation is a method designed to improve the performance of LLMs by selectively integrating external knowledge. The idea behind FLARE is to monitor the probabilities generated by the LLM during the generation of the answers. If the model generates a token with probability below threshold (i.e. the model is uncertain), FLARE intervenes by querying an external knowledge source, such as a search engine or structured knowledge base, to retrieve relevant information. Using this additional context, FLARE regenerates the response until the next uncertain token or ends the generation. This approach balances high-quality generation and high response speed.

DRAGIN (Su et al., 2024b) – **D**ynamic **R**etrieval **A**ugmented **G**eneration based on the **I**nformation **N**eeds of LLMs follows a similar approach to FLARE by monitoring the model's token probabilities during generation. If LLM produces tokens with low likelihood, indicating uncertainty or knowledge gaps, DRAGIN triggers a retrieval process. For better identification of uncertainty tokens, DRAGIN filters out all stopwords. [6]. This paper also introduces an additional step: reformulating the query with keywords before retrieving information. These reformulated keywords are based on the model's internal attention weights and reasoning, allowing the system to determine what information is necessary and target relevant external knowledge sources more effectively. By incorporating new knowledge and ensuring the relevance of the retrieved information, DRAGIN improves the coherence of the final response. This approach reduces the risk of retrieving irrelevant documents and optimizes the model's reasoning process, especially in situations where queries may be ambiguous or incomplete.

Rowen (Ding et al., 2024) – **R**etrieve **O**nly **W**hen It Needs method presents a novel approach to reducing hallucinations in LLMs. This method uses an adaptive retrieval mechanism to improve the accuracy of LLM output. The method intelligently determines when to use external knowledge sources, based on a language and model evaluation.

The Rowen Consistency Language (Rowen CL) component of Rowen involves generating semantically equivalent perturbations of the input query across English and Chinese languages. This includes asking the model to produce variations of the same question and then comparing the consistency of the responses generated in different languages. A high degree of inconsistency among these responses indicates uncertainty in the model's understanding, prompting the system to initiate a retrieval process to gather factual information that may clarify or correct the initial response. The Rowen Consistency Model (Rowen CM) extends this idea by assessing the semantic coherence of responses generated by dif-

[4] set to 8 in the experiments

[5] maximum amount of paragraphs is set to 15 to fit the model's context limit

[6] `https://spacy.io/usage/linguistic-features`

ferent models, OpenAI GPT-3.5 and Qwen-Max-0428[7], as described in the original paper. By comparing outputs from a primary language model with those generated by a verification model, final consistency model score calculated. Rowen Hybrid - the hybrid version of Rowen CL and Rowen-CM, if the sum of the consistency scores for both CL and CM is greater than the threshold, the retriever is used to mitigate potential hallucinations.

To ensure a reproducible and comparable evaluation of our work, we have reimplemented Rowen approach using LLaMA3.1-8b-instruct as the primary model and Qwen 2.5-72B-Instruct (Yang et al., 2024; Team, 2024) as the verification model for consistency model evaluation.

SeaKR (Yao et al., 2024) – **Se**lf-**a**ware **K**nowledge **R**etrieval for Adaptive RAG uses an uncertainty approach to minimise hallucinations in LLMs. SeaKR uses the model's internal states to extract self-aware uncertainty, activating external knowledge sources only when the LLM exhibits high uncertainty during generation. This selective retrieval mechanism increases the accuracy and reliability of the generated output.

The SeaKR Uncertainty Module (SeaKR UM) is a core component that monitors the internal states of the LLM to quantify its self-aware uncertainty. When the uncertainty level exceeds a predefined threshold, SeaKR UM triggers the retrieval process to retrieve relevant knowledge snippets from external databases. To ensure the most effective integration of the retrieved information, the SeaKR Re-ranking Component (SeaKR RC) re-orders the retrieved snippets based on their ability to reduce the model's uncertainty, selecting the snippet that provides the greatest clarity and factual accuracy.

To ensure a reproducible and comparable evaluation of our approach, we have reimplemented the SeaKR model using Llama-3.1-8b-instruct for the evaluation of self-conscious uncertainty. For consistency, we use the same eigenscore threshold as in the original paper because it gave the best results, but we have also tried others.

G Correlations between evaluation metrics across each dataset

	InAcc	EM	F1	Acc	AUC	Corr
InAcc	1.00	0.63	0.75	0.09	-0.02	0.05
EM	0.63	1.00	0.93	-0.12	0.09	0.09
F1	0.75	0.93	1.00	-0.06	0.08	0.09
Acc	0.09	-0.12	-0.06	1.00	0.21	0.15
AUC	-0.02	0.09	0.08	0.21	1.00	0.79
Corr	0.05	0.09	0.09	0.15	0.79	1.00

Table 14: Spearman correlations between evaluation metrics normalized across each dataset. The result reveals a low correlation between downstream metrics (InAcc, EM, F1) and self-knowledge metrics (Acc, AUC, Corr). This underscores the importance of conducting a more comprehensive evaluation of self-knowledge of adaptive retrieval systems, rather than relying solely on downstream performance.

H Retriever Evaluation

To assess the quality of the retrieval component, we conduct a comparative evaluation on two of our datasets: NQ and TriviaQA. We report standard retrieval metrics—`Recall@`k and `Mean Reciprocal Rank (MRR@`k`)`—which are most informative in our setting with a single ground-truth context per query. MAP is omitted, as it is equivalent to MRR in this case.

We compare our retriever choice, which is similar to DRAGIN and FLARE, against other baselines, including AdaptiveRAG, IRCoT, SeaKR, and ROWEN. These methods represent a mix of static and adaptive retrieval strategies with varying levels of model-query interaction.

[7] https://qwenlm.github.io/blog/qwen-max-0428/

Method acronym	Method full name	Short description
		logit based
FisherRao (Darrin et al., 2023)	Fisher-Rao distance	FisherRao is a distance on the Riemannian space formed by the parametric distributions, using the Fisher information matrix as its metric. It computes the geodesic distance between two discrete distributions.
Max Entropy (Fomicheva et al., 2020)	Maximum Token Entropy	The maximum entropy of all tokens in the generated sequence.
Max Probability	Maximum Sequence Probability	The score leverages the probability of the most likely sequence generation.
Mean CPMI (van der Poel et al., 2022)	Mean conditional pointwise mutual information	Extension of the PMI method by considering only those marginal probabilities for which the entropy of the conditional distribution is above certain threshold.
Mean Entropy (Fomicheva et al., 2020)	Mean Token Entropy	The average entropy of each individual token in the generated sequence.
Mean PMI (Takayama and Arase, 2019)	Mean pointwise mutual information	PMI compares the probability of two events (the question and the generated answer) occurring together to what this probability would be if the events were independent.
Mean Probability	Mean Sequence Probability	The total uncertainty is measured via average sequence probability.
Median Entropy (Fomicheva et al., 2020)	Median Token Entropy	The median entropy of all tokens in the generated sequence.
Median Probability	Median Sequence Probability	The total uncertainty is measured via median sequence probability.
Min Entropy (Fomicheva et al., 2020)	Minimum Token Entropy	The minimum entropy of all tokens in the generated sequence.
Min Probability	Minimum Sequence Probability	The score leverages the probability of the least likely sequence generation.
Perplexity (Fomicheva et al., 2020)	Perplexity	The score computes the average negative log probability of generated tokens, which is further exponentiated.
PTrue (Kadavath et al., 2022)	probability P(true)	The method measures the uncertainty of the claim by asking the LLM itself whether the generated claim is true or not. The confidence is the probability of thefirst generated token y1 being equal to "True".
RenyiNeg (Darrin et al., 2023)	Rényi negentropy	The score computes alpha-Renyi-divergence between the sample and the uniform distributions.
SAR (Duan et al., 2023)	Shifting Attention to more Relevant	SAR corrects generative inequalities by reviewing the relevance of each token and emphasizing uncertainty quantification attention to those more relevant components. The relevance is measured by calculating similarity of sentence before and after removing the certain token.
SentenceSAR (Duan et al., 2023)	Shifting Attention to more Relevant at Sentence level	SAR measured at sentence-level.
		consistency based
CCP (Fadeeva et al., 2024a)	Claim-Conditioned Probability	The method aggregates token-level uncertainties into a claim-level score, it removes the impact of uncertainty about what claim to generate on the current step and what surface form to use.
DegMat (Lin et al., 2023)	Degree matrix	Using the Degree matrix a new uncertainty measure could be found that reflects the average pairwise distance.
Eccentricity (Lin et al., 2023)	Eccentricity	The smallest k eigenvectors of Laplacian Graph are used as the proxy for the models' embeddings. Then, we could use the average offset from the average embedding as the uncertainty measure.
EigValLaplacian (Lin et al., 2023)	Sum of eigenvalues of the graph Laplacian	The score uses pairwise similarities between the sampled answers to the questions to form the symmetric weighted adjacency matrix (degree matrix). This matrix is further used to create the graph Laplacian. The sum of Eigenvalues of the Graph Laplacian are used as a measure of uncertainty.
Lex-Similarity (Fomicheva et al., 2020)	Lexical similarity	The score computes how similar two words or phrases are in terms of their meaning.
NumSemSets (Lin et al., 2023)	Number of semantic sets	The number of semantic sets initially equals the total number of generated answers K. If two answers are semantically similar, they are put into one cluster. A higher number of semantic sets corresponds to an increased level of uncertainty, as it suggests a higher number of diverse semantic interpretations for the answer.
Semantic Entropy (Kuhn et al., 2023)	Semantic Entropy	The method aims to deal with the generated sequences that have similar meaning while having different probabilities according to the model. The idea is to cluster generated sequences into several semantically homogeneous clusters with a bi-directional entailment algorithm and average the sequence probabilities within the clusters.
		internal-based
MD (Lee et al., 2018)	Mahalanobis distance	In this paper, the authors propose a simple yet effective method for detecting any abnormal samples, which is applicable to any pre-trained softmax neural classifier. They obtain the class conditional Gaussian distributions with respect to (low- and upper-level) features of the deep modelsunder Gaussian discriminant analysis, which result in a confidence score based on the Mahalanobis distance.
RDE (Yoo et al., 2022)	Robust density estimation	The method improves over MD by reducing the dimensionality of the last hidden state of the decoder averaged over all generated tokens via PCA decomposition. Additionally, computing of the covariance matrix for each individual class is done by using the Minimum Covariance Determinant estimation. The uncertainty score is computed as the MD in the space of reduced dimensionality.
RMD (Ren et al., 2023)	Relative Mahalanobis distance	The MD distance score is adjusted by subtracting from it the other MD score computed for some large general purpose dataset covering many domains.
		blended approach
Hybrid	Hybrid	Our hybrid approach that uses all uncertainty features defined in the table.

Table 15: Description of the uncertainty estimation methods used in the paper. The methods are grouped by their categories: logit based, consistency-based, internal-based and hybrid.

TriviaQA. Table 17 shows results on TriviaQA. Our retriever achieves strong performance, particularly in terms of MRR, outperforming all baselines at lower ranks and maintaining competitive scores across all settings.

Method	NQ			SQUAD			TQA			2Wiki			HotPot			Musique		
	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓	InAcc ↑	LMC ↓	RC ↓
Never RAG	0.446	1.0	0.00	0.176	1.0	0.00	0.636	1.0	0.00	0.318	1.0	0.00	0.286	1.0	0.00	0.106	1.0	0.00
Always RAG	0.496	1.0	1.00	0.312	1.0	1.00	0.610	1.0	1.00	0.374	1.0	1.00	0.410	1.0	1.00	0.100	1.0	1.00
AdaptiveRAG	**0.496**	2.0	0.98	0.286	2.0	0.97	0.636	1.5	0.54	0.454	5.2	2.64	**0.44**	4.7	2.41	**0.154**	3.6	3.63
DRAGIN	0.480	4.5	2.24	**0.298**	4.3	2.14	**0.667**	4.0	2.0	**0.456**	5.8	2.92	0.435	5.1	2.5	0.134	6.3	3.15
FLARE	0.462	4.26	2.0	0.288	3.2	2.5	0.648	2.1	1.39	0.424	3.9	2.85	0.372	5.1	4.07	0.106	4.3	3.11
Seakr	0.406	14.6	1.00	0.286	14.0	1.00	0.656	14.6	1.00	0.398	12.3	2.44	0.424	9.9	1.76	0.118	12.3	2.40
Ideal	0.608	1.6	0.55	0.360	1.8	0.82	0.736	1.4	0.36	0.500	1.7	0.68	0.460	1.7	0.71	0.164	1.9	0.89

Table 16: QA Performance of adaptive retrieval fine-tuned with in-domain data. 'Ideal' represents the performance of a system with an oracle providing ideal predictions for the need to retrieve. 'InAcc' denotes In-Accuracy, measuring the QA system's performance. 'LMC' indicates the mean number of LM calls per question, and 'RC' represents the mean number of retrieval calls per question. AdaptiveRAG and DRAGIN methods show the best performance.

Metric	**Ours**	**AdaptiveRAG**	**IRCoT**	**SeaKR**	**ROWEN**
Recall@3	0.417	0.395	0.412	0.419	0.414
Recall@5	0.490	0.458	0.480	0.491	0.485
Recall@10	0.559	0.538	0.524	0.568	0.572
MRR@3	0.341	0.307	0.323	0.329	0.332
MRR@5	0.358	0.322	0.338	0.345	0.348
MRR@10	0.367	0.332	0.344	0.357	0.359

Table 17: Retriever performance on TriviaQA.

Natural Questions. Results for the NQ dataset are presented in Table 18. Our method performs competitively across all metrics, particularly at lower ranks, and surpasses AdaptiveRAG and IRCoT in early precision. SeaKR and ROWEN achieve stronger results at higher k, likely due to more compute applied to preprocess queries and reformulate every sentence.

Metric	Ours	AdaptiveRAG	IRCoT	SeaKR	ROWEN
Recall@3	0.253	0.254	0.252	0.342	0.358
Recall@5	0.326	0.305	0.306	0.407	0.438
Recall@10	0.389	0.371	0.438	0.508	0.536
MRR@3	0.186	0.186	0.184	0.257	0.283
MRR@5	0.203	0.198	0.197	0.278	0.302
MRR@10	0.211	0.206	0.214	0.301	0.315

Table 18: Retriever performance on Natural Questions.