Quantifying the Robustness of Retrieval-Augmented Language Models Against Spurious Features in Grounding Data

Anonymous ACL submission

Abstract

Robustness has become a critical attribute for the deployment of RAG systems in real-world applications. Existing research focuses on robustness to explicit noise (e.g., document semantics) but overlooks implicit noise (spurious features). Moreover, previous studies on spurious features in LLMs are limited to specific types (e.g., formats) and narrow scenarios (e.g., ICL). In this work, we statistically demonstrate the presence of spurious features in the RAG paradigm, a robustness problem caused by the sensitivity of LLMs to semantic-agnostic features. Then, we propose a comprehensive taxonomy of spurious features and empirically quantify their impact through controlled experiments. Our analysis reveals that not all spurious features are harmful and they can even be beneficial sometimes. Further evaluation results suggest that spurious features are a widespread and challenging problem in the field of RAG. The code and dataset will be released to facilitate future research.

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as a promising paradigm to mitigate LLMs hallucinations (Gao et al., 2023; Yang et al., 2023a), integrating relevant external knowledge to improve the factuality and trustworthiness of LLM-generated outputs (Zhou et al., 2024). However, Retrieval-Augmented Language Models (RALMs) still face substantial robustness issue due to the presence of noise in retrieved documents (Liu et al., 2023; Li et al., 2024b).

Recent research aims to explore the characteristics that affect the robustness of RAG systems from the perspective of grounding data construction (Cuconasu et al., 2024). These studies examine various factors, including the type (Wu et al., 2024a), number (Xu et al., 2024), and position of documents (Liu et al., 2024) within the prompt context. However, previous analyses primarily focus on explicit noise that significantly alter the semantic information (causal features) of grounding data(Wu et al., 2024b; Cuconasu et al., 2024), while neglecting implicit noise (spurious features) that introduce semantic-agnostic modifications. This limitation extends to existing evaluation benchmarks, which simulate complex noise scenarios to assess the robustness of RAG systems (Chen et al., 2024a; Wu et al., 2024a), yet lack available benchmarks and metrics to measure the robustness of RALMs against spurious features. A more detailed discussion of related work is in Appendix A.1.

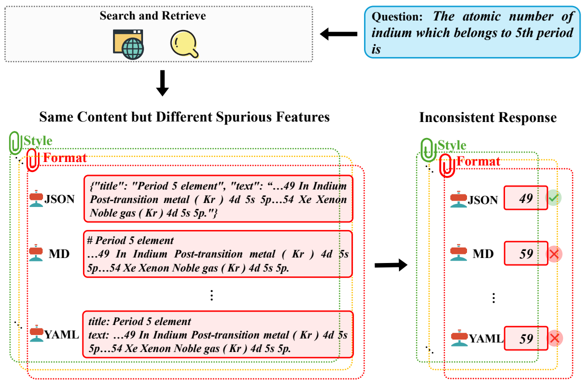


Figure 1: An example from the *SURE_Wiki* dataset (Sec. 4), illustrating the sensitivity of RAG systems to spurious features within grounding data. The original retrieved document is fed into the LLMs in different formats, leading to inconsistent responses.

Contemporary RAG systems typically employ production-level retrievers, such as Bing and Google, to retrieve relevant information from the internet. Unlike a single corpus, the internet encompasses diverse data with distinct features. For any given query, there may exist numerous golden documents that contain the correct answer but differ in style, format, or other attributes. As shown in Figure 1, we have observed that LLMs may fail to consistently derive the correct answer from golden documents with different formats. A similar phenomenon is reported in Sclar et al. (2024) and He et al. (2024), which demonstrate that LLMs are extremely sensitive to the format of prompts (i.e., spurious features). For more related work, see Appendix A.2. Unfortunately, there is no statistic and empirical evidence to support the existence of spurious features in the RAG paradigm. This highlights the urgent need to redefine spurious features in RAG and systematically quantify the robustness of RALMs against them.

To address these challenges, we first design a preliminary experiment to demonstrate that RALMs are sensitive to semantic-agnostic features in the grounding data, thereby extending the definition of spurious features to RAG systems. Building on findings from our preliminary experiment and recent studies, we identify five common types of spurious features in RAG scenarios. Then, we propose a novel framework, *SURE*, for automating the process of robustness evaluation. This framework follows a *perturb-then-evaluate* approach, offering great scalability. In *SURE*, automated perturbations are applied to the original instances to inject the corresponding spurious features. The perturbed instances are then examined to ensure that the causal features remain intact. After these steps, we employ tailored metrics to quantify the robustness of RALMs against spurious features. Further analysis reveals that not every spurious features is harmful and they can even be beneficial sometimes. To enable more efficient evaluation, we distill the most challenging instances from the synthetic data generated by our framework to create a lighter benchmark, *SIG*. Extensive evaluations on diverse LLMs and methods indicate that maintaining robustness against spurious features is a widespread challenge.

Our contribution can be summarized as follows: **1)** We identify and define spurious features in RAG systems. To the best of our knowledge, this is the first comprehensive study to define and evaluate spurious features from RAG perspective. **2)** We propose a novel evaluation framework, *SURE*, to assess the robustness of RALMs against spurious features, which includes a comprehensive taxonomy, tailored metrics, and a data synthesis pipeline. **3)** We curate a lightweight yet challenging evaluation dataset, and offer valuable insights and baselines for future research through extensive experiments and analysis.

2 Preliminary

In this section, we first define causal and spurious features in the context of RAG and then demonstrate the existence of spurious features statistically.

2.1 Causal and Spurious Features in RAG

In general, causal features are input features that have a direct causal effect on the output of predictive model (Yu et al., 2020). Their relationship is rooted in causality, rather than mere statistical correlation. When it comes to Large Language Models, the meaning and intent of prompts serve as causal features that directly influence the models' responses. In the context of RAG, causal features refer to the semantic information of grounding data.

In contrast, spurious features are input features that co-occur with causal features and are erroneously captured by the model (Neuhaus et al., 2023). These features exhibit a statistical correlation with the model's output but lack a causal relationship. Recent research has shown that LLMs are sensitive to seemingly trivial features like prompt formatting, thereby extending the definition of spurious features to LLMs (Sclar et al., 2024). Similarly, we define the semantic-agnostic features of the grounding data as spurious features in RAG systems. However, conclusions drawn from in-context learning scenarios (e.g., classification and multiple-choice tasks) may not applicable to RAG scenarios, which typically involve open-ended generation tasks. Therefore, we design a preliminary experiment to validate the presence of spurious features in RAG.

2.2 Preliminary Experiment

We aim to demonstrate the semantic-agnostic features within real documents are spurious features, i.e., to reveal their impact on the output of RAG systems.

There are some challenges in revealing the influence of semantic-agnostic features. First, when retrieving from a single corpus, it is difficult to mine semantically equivalent counterparts that differ only in semantic-agnostic features. To mine appropriate documents, we introduce *Contriever-msmarco*, a traditional dense retriever, to recall 100 semantically similar candidates. To further eliminate the effect of causal features, documents without golden answers are filtered out, ensuring that the remaining documents have roughly consistent causal features.

Still, the differences in spurious features among these candidate documents are often minor, and their impact on model responses cannot be effectively captured using binary evaluation methods that simply judge whether an answer is correct or incorrect. Thus, more fine-grained metrics are required to detect such nuanced performance changes. Inspired by the use of LLMs as supervision signals for document utility (Izacard et al., 2023; Gan et al., 2024), we introduce the *oracle score*, which measures fine-grained performance through calculating the log probability of generating correct answers given a specific document. The *oracle score* is defined as follows:

$$\mathrm{Oracle}(x, y, \theta) = \sum_{t=1}^{T} \log p(y_t \mid x, y_{<t}, \theta) \quad (1)$$

where x is the input prompt for RALMs, including the instruction I, grounding data G, and query Q; y represents the ground truth answer; θ denotes the model parameters; and T is the total length of the answer sequence [1].

For each query, we construct document pairs by selecting the first-ranked and last-ranked candidate documents based on their oracle scores. However, the presence of various semantic-agnostic features within each document pairs makes it challenging to isolate the impact of any individual features. To assess the influence of a given feature, we compare its distribution between document sets with first- and last-ranked oracle scores. A control group is constructed by randomly sampling two document sets. If the distributions differ significantly, it suggests that RALMs are sensitive to the feature. See Appendix B for implementation details of preliminary experiments.

We test the following features: 1) Flesh Score, 2) Distinct-1, 3) Dependency Tree Depth, 4) PPL, and 5) Token Length. The results show that RALMs are sensitive to semantic-agnostic features. Nevertheless, it does not offer empirical evidence or quantitative analysis. Inspired by previous data synthesis studies (Tan et al., 2024b), we use a data synthesis approach to better control feature variables and quantify the robustness of RALMs.

3 Proposed Framework

In this section, we detail our proposed evaluation framework, *SURE* (**S**purious Feat**U**res **R**obustness **E**valuation), which designed specifically for assessing the robustness of RALMs against spurious features in grounding data. As illustrated in Figure 2, this framework comprise four components: **1)** *Comprehensive Taxonomy.* We identify and define five common types of spurious features in RAG scenarios. **2)** *Spurious Features Injection.* We design a data synthesis pipeline to automate the injection of spurious features, utilizing both model-based and rule-based methods to construct counterparts of the original document with varying spurious features. **3)** *Causal Features Preservation.* We employ a bidirectional entailment algorithm and a string matching strategy to ensure that the causal features of grounding data remain unchanged. **4)** *Robustness Evaluation.* We introduce three metrics (Win Rate, Lose Rate, and Robustness Rate) to facilitate fine-grained, instance-level evaluation.

3.1 Problem Formulation

Given a query q , the retriever R returns a list of relevant documents from a corpus $D = \{d_i\}_{i=1}^{N}$. The relevance between document d and query q can be measured by various methods. In this work, we use a BERT-based dense retriever to obtain the embedding of query and documents, respectively. The relevance score is calculated by computing their dot-product similarity. Then, the Top-k documents with the highest similarity scores are retrieved:

$$D_{\text{retrieve}} = \text{argtop-}k \left\{ s(q, d_i) \mid d_i \in D \right\}. \quad (2)$$

To formally quantify the robustness of RAG systems against spurious features, we define the input prompt for the LLM-based reader as $P = (I, G, Q)$, where I represents instruction, G refers to the grounding data, constituted by a subset of D_{retrieve}, and Q is the query. A perturbation is introduced to investigate the impact of spurious features by applying a semantic-agnostic modification to the original grounding data, while preserving its causal features. We define $g(.)$ to automate this process, transforming G to $g(G)$ and producing a counterpart $\hat{P} = (I, g(G), Q)$. The outputs of LLM-based reader for P and $\hat{P}$ are compared to evaluate the impact of the introduced perturbation:

$$y = \text{LLM}(P), \quad \hat{y} = \text{LLM}(\hat{P}). \quad (3)$$

3.2 Taxonomy of Spurious Features

We develop a comprehensive taxonomy of spurious features, informed by our preliminary experiments and insights from prior research. The five types of

[1]For cases with multiple answers, we compute the final score by averaging the corresponding oracle scores across all answers.

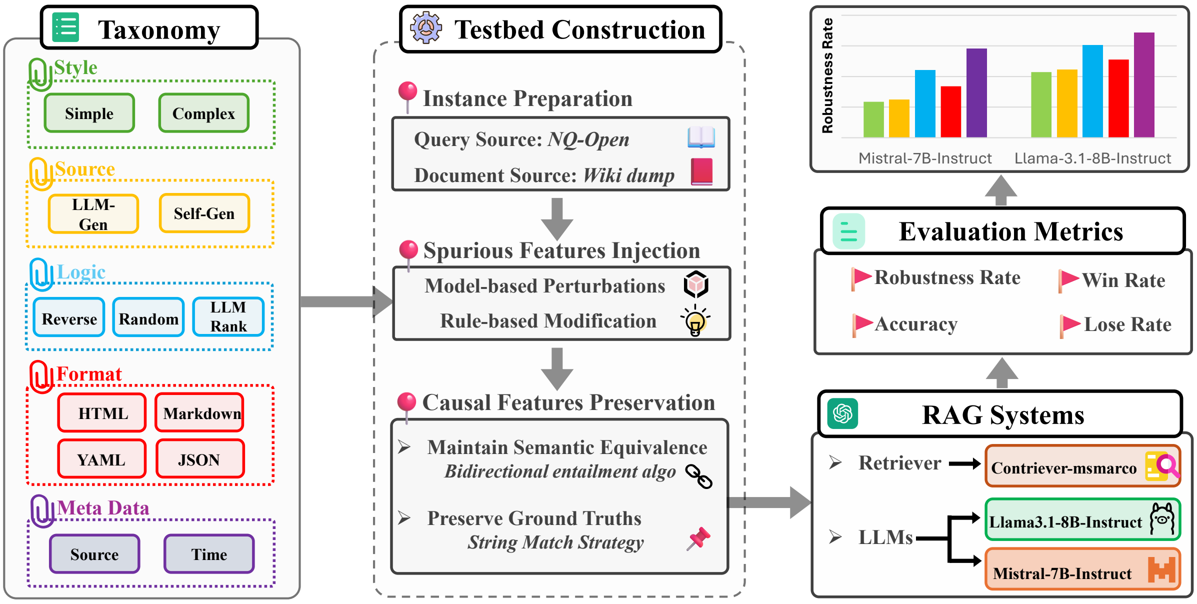


Figure 2: Overview of our SURE framework. We provide a *Comprehensive Taxonomy* that includes five types of spurious features, further divided into 13 subtypes of perturbations (left section). To construct the testbed, we prepare raw instances initially and then synthesize the modified instances through a workflow consisting of *Spurious Features Injection* and *Causal Features Preservation* (middle section). By applying carefully tailored metrics for *Robustness Evaluation*, we quantify the robustness of target RAG systems (right section).

spurious features and their corresponding perturbations are detailed below.

Style Perturbations The same content can be expressed in different styles, using varying tones, words and sentence structures. As shown in Section 2.2, LLMs exhibit biases towards readability-related features. Similarly, for humans, the readability of a text can significantly influence its accessibility to the audience (Yang et al., 2023b). Therefore, we define two perturbations from the perspective of readability style: **Simple** and **Complex**. The former simplifies the grounding data by using basic vocabulary and simple sentence structure, while the latter employs professional vocabulary and a formal academic tone to complex the documents.

Source Perturbations LLM-generated content, including both misinformation and correct claims, infiltrates every corner of the internet. Recent studies have shown that neural retrievers are biased towards LLM-generated content, leading to the marginalization of human-authored content (Dai et al., 2024; Chen et al., 2024b). Moreover, our preliminary experiments demonstrate that LLMs are biased towards the Perplexity (PPL) of text. Thus, we define two types of source perturbations: **LLM-generated** and **Self-generated**. Specifically, the LLM-generated perturbation paraphrases the original document using a powerful LLM, while the self-generated perturbation employs the same backbone model used as the generator in the RAG system.

Logic Perturbations In RAG systems, documents are often segmented into multiple chunks and may be retrieved in varying orders. Here, we simulate scenarios where the intrinsic logical flaw is disrupted by three different perturbations: **Random**, **Reverse**, and **LLM-reranked**, each representing a distinct sentence ordering strategy.

Format Perturbations The internet contains various data formats, including **HTML**, **Markdown**, **YAML** and **JSON**. These formats are usually processed into plain text before being fed to LLMs. To mitigate the loss of structural information during this process, some RAG studies propose using the original format, rather than plain text, to augment the generation (Tan et al., 2024a). However, as highlighted in previous research, the prompt format is recognized as a spurious feature that can significantly impact model performance (Sclar et al., 2024; He et al., 2024). Therefore, we perturb the original document with four common formats to explore the impact of grounding data format in the context of RAG.

Metadata Perturbations Metadata is often included in the HTML results returned by search engines. In our framework, we focus on two types: **Timestamp** and **Data source**. The timestamp marks when the data was created, and the data source indicates its origin. For timestamp perturbations, *pre* and *post* denote whether the timestamp is before or after the LLM's knowledge cutoff date. For data source perturbations, *wiki* and *twitter* represent the domains of the URLs.

3.3 Spurious Features Injection

The automation of spurious features injection is essential for automating the entire evaluation framework. We detail the process of collecting the original instances and describe how the automated perturbation was implemented.

Instance Preparation An instance is the dynamic component of the prompt P, consisting of a query Q and grounding data G. To construct the original instances, we first select 1,000 queries from the NQ-open dataset. For each query, we then retrieve 100 documents from the Wikipedia dump to serve as grounding data, yielding 100,000 instances for the following perturbation step.

Automated Perturbation As introduced in Section 3.1, the perturbation $g(.)$ injects spurious features by modifying the grounding data. For style and source perturbations, $g(.)$ is implemented using an LLM[2] prompted by carefully crafted guidelines to modify the raw document, producing counterparts of the original instances. For logic and format perturbations, we develop $g(.)$ as a heuristic method based on a set of predefined rules[3]. To simulate real-world metadata, we first synthesize pseudo Wikipedia or Twitter links for the raw instances, and then organize them into HTML format using a rule-based $g(.)$. The complete implementation details for automated perturbation are provided in Appendix C.

3.4 Causal Features Preservation

To eliminate the effect of causal features, it is essential to follow the principle of controlled experiments by keeping causal features constant while systematically manipulating spurious features. This approach isolates the impact of spurious features from that of causal features, enabling an accurate quantification of robustness against spurious features. In our framework, we introduce two methods to ensure the stability of causal features in the grounding data. Implementation details can be found in Appendix D.

Maintain Semantic Equivalence For models capable of following human instructions, we directly instruct them to maintain semantic equivalence when injecting spurious features. Nonetheless, it's impossible to completely avoid semantic shift during the perturbation process. To ensure the semantic consistency before and after introducing perturbation, we employ a bidirectional entailment algorithm to filter out instance pairs (raw instance, perturbed instance) with semantic inequivalence. Specifically, for document G and its modified counterpart $g(G)$, we use a Natural Language Inference (NLI) system to detect whether the latter can be inferred from the former, and vice versa. The NLI system classifies predictions into one of: *entailment*, *neutral*, *contradiction*. We compute both directions, and the algorithm returns *equivalent* if and only if both directions are predicted as entailment.

In general, this algorithm can be implemented by any NLI system. However, in our case, the concatenation of G and $g(G)$ sometimes exceeds the context limitation of a Bert-based NLI model. Hence, we apply an LLM-based NLI system [4] to implement the bidirectional entailment algorithm.

Preserve Ground Truths While semantic equivalence protects causal features to the greatest extent, the perturbation may lead to the correct answer being paraphrased into an alias (e.g., "President Roosevelt" to "Roosevelt"). These variations in the grounding data are likely to result in false negatives when determining response correctness, despite the NQ-Open dataset providing multiple potential answer variants for each query. To address this issue, we employ a simple string-matching strategy to filter out documents that have undergone unexpected modifications.

3.5 Robustness Evaluation

We employ an evaluation method $Y(.)$, in line with Liu et al. (2024); Cuconasu et al. (2024),

[2]Unless otherwise specified, all model-based $g(.)$ are implemented using *Llama-3.1-70B-Instruct*.

[3]One exception is that we implement the LLM-reranked perturbation using an LLM-based $g(.)$.

[4]Farquhar et al. (2024) confirms the effectiveness of the LLM-based NLI system through human annotation, demonstrating that its performance is on par with the DeBERTa-large model used in Kuhn et al. (2023).

to measure the correctness of responses generated by RAG systems. This approach checks whether any of the correct answers is contained within the response produced by the LLM and then derives a binary label. Previous researches use accuracy as the primary metric and report it at dataset level to assess the robustness of RALMs, which is quantified by calculating the variations in the models' accuracy across different types of noise. However, dataset-level metrics has certain limitations, as it may fail to capture fine-grained variations that occur at the instance level. As shown in Figure 3, RALMs may appear robust at dataset-level evaluations but exhibit significant sensitivity at the instance level.

To quantify whether a RAG system is robust and unbiased at the instance level, we assign a ternary label to each instance by comparing the correctness of the LLM's response before and after introducing the perturbation. This comparison process can be formulated as $C = Y(y_i) - Y(\hat{y}_i)$, where C lies in the set $(-1, 0, 1)$. Based on the comparison outcomes, we define three metrics: **Robustness Rate (RR)**, **Win Rate (WR)**, and **Lose Rate (LR)**. The RR is calculated as follows:

$$\mathrm{RR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(C == 0) \tag{4}$$

where N is the total number of instances in the dataset; y_i and $\hat{y}_i$ represent the outputs of LLM for the original and perturbed instances. RR measures the proportion of instances where the RALM's answer remains consistent (0) before and after introducing the perturbation. Similarly, WR and LR quantify the proportions of instances where the correctness of the RALM's response changes after the perturbation, either from incorrect to correct ($C == -1$) or from correct to incorrect ($C == 1$).

4 Experiments

In this section, we assess the robustness of RAG systems to spurious features by evaluating them on their most popular application—the Question Answering (QA) task, following the standard "retrieve-read" setting of the RAG paradigm.

4.1 Experimental Setup

Datasets Through the steps of **spurious features injection** and **causal features preservation**, we derive the final dataset available for robustness evaluation: *SURE_Wiki*. The queries are drawn from

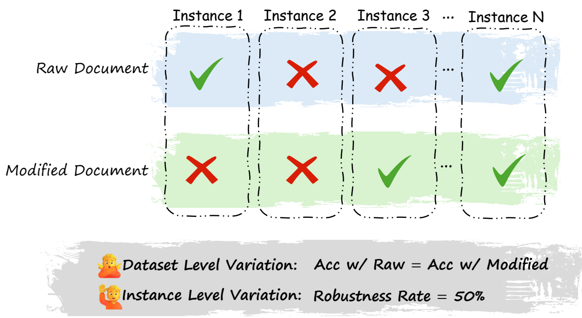


Figure 3: A comparison of dataset-level metric (Acc) and instance-level metric (RR) for robustness evaluation. ✔ and ✗ indicate the correctness of responses. In this example, RR captures instance-level unrobustness, while Acc overlooks RALMs' sensitivity to spurious features within documents.

the NQ-open dataset (Lee et al., 2019), while our data source is English Wikipedia dump.

Models We test two representative LLMs in our main experiments: *Mistral-7B-Instruct-v0.3* and *Llama-3.1-8B-Instruct*. Further implementation details are included in Appendix F.

4.2 Result Analysis

To further analyze spurious features, we divide *SURE_Wiki* into four subsets based on the categories of queries and documents within each instance. A query is labeled as *Known* if it can be correctly answered in a closed-book setting; otherwise, it is labeled as *Unknown*. Documents are categorized as *Golden* or *Noise* depending on whether they contain ground truths. Notably, the distribution of the dataset is model-specific, as the classification of *Known* and *Unknown* queries is determined by the intrinsic knowledge of the target LLM. Table 2 presents dataset statistics for *Mistral-7B-Instruct-v0.3*, while the distribution for *Llama-3.1-8B-Instruct* is shown in Appendix E.

For Different Queries and Grounding Data We report the results of *Mistral-7B-Instruct* and *Llama-3.1-8B-Instruct* in Table 1 and Table 7, respectively. For golden documents, the robustness rates of K-G and U-G are very similar for both *Mistral* and *Llama*, whereas their accuracy differ significantly. This suggests that, unlike robustness to explicit noise (Wu et al., 2024b), **robustness against spurious features is independent of the model's internal prior knowledge**.

When tested on noise documents, the RR remains high across all spurious features, as LLMs

Mistral-7B-Instruct-v0.3																		
Taxonomy	**Perturbations**	**Known-Golden**					**Known-Noise**					**Unknown-Golden**					**U-N**
		LR	RR	WR	Org	Acc	LR	RR	WR	Org	Acc	LR	RR	WR	Org	Acc	RR
Style	Simple	7.33	85.00	**7.67**	73.02	73.37	4.45	91.64	3.90	10.82	10.28	7.87	82.95	**9.18**	56.31	57.62	98.76
	Complex	6.05	87.42	**6.53**		73.50	3.85	92.03	**4.12**		11.10	6.90	85.92	**7.17**		56.58	98.82
Source	LLM-Generated	5.91	87.62	**6.47**	71.81	72.36	3.57	92.27	**4.16**	10.79	11.38	6.41	86.52	**7.06**	54.46	55.11	98.75
	Self-Generated	6.30	87.06	**6.64**		72.15	3.94	92.02	**4.04**		10.89	6.26	86.80	**6.94**		55.14	98.77
Logic	Reverse	5.44	89.34	5.22	69.91	69.69	2.99	94.10	2.92	11.77	11.70	5.97	88.54	5.49	50.26	49.79	99.04
	Random	4.47	91.87	3.66		69.10	2.43	95.15	2.42		11.76	4.18	91.44	**4.38**		50.46	99.27
	LLM-Ranked	3.52	93.15	3.33		69.72	2.07	95.84	**2.09**		11.79	3.57	92.89	3.54		50.24	99.30
Format	JSON	7.96	88.53	3.51	70.81	66.35	5.15	92.68	2.17	10.98	8.00	6.95	88.92	4.13	53.32	50.50	99.02
	HTML	9.30	87.03	3.67		65.18	5.89	92.36	1.74		6.83	8.36	87.39	4.25		49.22	99.01
	YAML	4.75	90.90	4.35		70.41	3.88	93.24	2.87		9.97	5.05	90.53	4.42		52.69	99.06
	Markdown	3.98	92.49	3.53		70.36	2.91	94.36	2.72		10.79	4.11	92.59	3.31		52.52	99.15
Metadata	Timestamp (pre)	2.62	94.90	2.48	65.04	64.90	1.28	97.61	1.11	6.83	6.66	3.15	94.45	2.40	48.08	47.33	99.67
	Timestamp (post)	2.74	94.87	2.40		64.70	1.16	97.63	**1.21**		6.88	3.45	94.41	2.14		46.77	99.68
	Datasource (wiki)	3.78	92.31	**3.91**		65.17	1.5	96.66	**1.84**		7.16	3.69	92.95	3.36		47.76	99.48
	Datasource (twitter)	2.68	93.59	**3.73**		66.08	1.3	97.22	**1.48**		7.00	2.04	94.90	**3.06**		49.10	99.59

Table 1: Robustness evaluation results of *Mistral-7B-Instruct-v0.3* on the *SURE_Wiki* dataset. *Org* indicates the accuracy on original instances, while *Acc* refers to the accuracy after introducing perturbations. We use **Bold** to mark the WR values that are higher than the LR, suggesting that the perturbation is beneficial.

	K-G	**K-N**	**U-G**	**U-N**	**Total**
Style	7766	31152	2593	37692	79203
Source	9249	32435	3228	39101	84013
Logic	9724	35537	3587	41990	90838
Format	11037	38018	4141	45518	98714
Meta	11104	38018	4255	45420	98797

Table 2: Statistics of the *SURE_Wiki* dataset for *Mistral-7B-Instruct-v0.3*. K-G denotes the instances composed of (*Known* query, *Golden* Document), while U-N refers to the instances consisting of (*Unknown* query, *Noise* Document). The values represents the number of instance pairs for each type of perturbations within the category of spurious features.

consistently generate incorrect responses in the absence of ground truths. In this case, even though the responses change, the RR does not decrease since all responses remain incorrect. This stems from the evaluation method of the proposed RR metric, which measures unrobustness by tracking changes in answer correctness rather than minor variations in responses. This design focuses on meaningful differences in user-relevant performance. Therefore, we primarily focus on the RR results for the golden documents in the following experiments.

For Different Perturbations We observe notable differences in robustness rates across the five types of spurious features. However, within each category, the RR values for different perturbations are relatively similar. Hence, the robustness of spurious features can be estimated by averaging the RR values of their corresponding perturbations.

When further comparing perturbations within the same category, we find that while their RR values are comparable, their WR and LR can differ significantly. If the WR exceeds the LR, more instances are corrected than misanswered after introducing perturbations. This suggests that **not every spurious feature is harmful and they can even be beneficial sometimes**.

4.3 SIG Benchmark & Further Analysis

The raw synthetic dataset is not ideal for extensive evaluation due to its large size. Furthermore, the class imbalance result in unfair comparisons across different types of spurious features. To facilitate more efficient evaluation, we extract the most challenging data from our synthetic datasets to create a lightweight benchmark: *SIG* (**S**purious features **I**n **G**olden document) [5].

Are Spurious Features a Widespread Problem? To examine whether spurious features are merely artifacts of specific model choices, we evaluate a diverse set of SOTA LLMs on the *SIG* benchmark. The evaluated models include *GPT-4O*, *GPT-4O-mini*, *Mistral-Large-Instruct* [6], *Llama-3.3-70B-Instruct*, *Qwen2.5-72B-Instruct*, and *DeepSeek-V3* (671B,MoE), covering a wide range of model series and architectures. To better compare the robustness of different models, we average the RR of each

[5] Specifically, we randomly select 100 instance pairs for each perturbation where both models lack robustness.

[6] https://huggingface.co/mistralai/Mistral-Large-Instruct-2411

perturbation within a category to derive the overall robustness for a specific type of spurious feature. The performance of six SOTA LLMs is then visualized using a radar chart, as shown in Figure 4. Despite the impressive robustness of closed-source models, they still exhibit sensitivity to certain specific perturbations. **These results demonstrate that spurious features are a widespread issue across different model families, sizes, and architectures (Dense VS. MoE).**

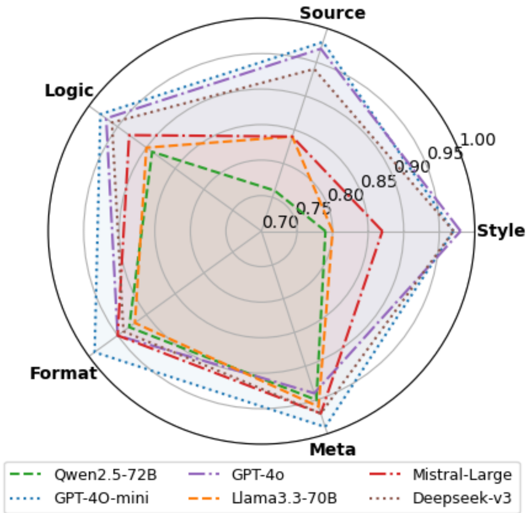


Figure 4: Robustness comparison of six SOTA LLMs.

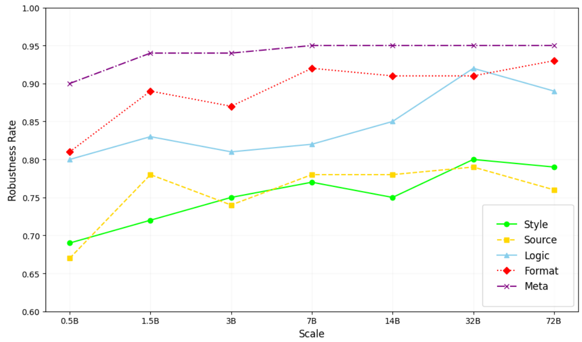


Figure 5: Scaling analysis on Qwen2.5 series.

Can Scaling up Model Size Solve the Problem? To investigate the impact of parameter scale on RAG robustness, we gradually increase the size of LLM-based readers (Qwen2.5 series, ranging from 0.5B to 72B) and evaluate their robustness across five types of spurious features. As illustrated in Figure 5, the robustness rate for all spurious features shows a relatively upward trend as the model size increases. However, when we further scale the model from 32B to 72B, the RR undergoes a significant decline (except for format and meta). Interestingly, for meta perturbations, while RALMs demonstrate strong robustness across all scales, their performance receives little to no benefit from scaling up. These findings suggest that although scaling up model size can enhance robustness to some extent, it fails to fundamentally eliminate sensitivity to spurious features.

Are Existing Robustness Solutions Effective? We evaluate whether methods developed to improve the robustness of RALMs against explicit noise can generalize to spurious features. Previous work, such as Chain-of-Note (CON) (Yu et al., 2023), aims to enhance robustness by generating thorough rationale before producing the answer. Moreover, recent breakthroughs in the reasoning capabilities of LLMs have significantly advanced the cutting edge of RAG. By integrating with reasoning models, RAG can overcome previous limitations and adapt to more complex scenarios (Gao et al., 2025). Therefore, we test both CON and DeepSeek-R1 on our SIG benchmark. Notably, the robustness rate of CON is even lower than the baseline without applying CON. A similar phenomenon was observed in experiments with the reasoning model DeepSeek-R1 (Guo et al., 2025), whose robustness was even worse than its base model, DeepSeek-V3. **This indicates that the robustness against spurious features cannot be effectively improved through COT-style techniques**.

	Style	Source	Logic	Format	Meta
Qwen2.5-72B	78.5	76.0	88.6	92.5	95.0
+ Chain-of-Note	74.0	**81.7**	66.7	84.8	91.0
DeepSeek-V3	96.5	93.6	95.6	94.0	96.5
DeepSeek-R1	84.5	87.3	83.3	87.0	87.5

Table 3: Robustness evaluation of CoN and DeepSeek-R1. Values that show improvements over the baseline are marked in **bold**.

5 Conclusion

In this work, we formally highlight the spurious features problem in RAG system. Through preliminary experiments, we provide statistical evidence to support the presence of spurious features in RALMs. We also propose a novel evaluation framework, *SURE*, to assess the robustness of RALMs against spurious features. Extensive evaluations and in-depth analysis highlight the urgent need to develop solutions for addressing spurious features in RAG systems.

6 Limitations

We strive to comprehensively cover all types of spurious features that may arise in RAG scenarios. However, some unidentifiable spurious features may fall outside the scope of our taxonomy and thus fail to be quantified using the proposed *SURE* framework. Furthermore, while our experiments highlight the limitations of existing RAG robustness solutions in addressing spurious features, we do not propose effective methods to enhance the robustness of RALMs against them.

7 Ethics Statement

We construct our testbed using publicly available seed data. During the data synthesis process, we carefully preserve the original semantics, thereby avoiding the generation of toxic content.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Ning Bian, Hongyu Lin, Peilin Liu, Yaojie Lu, Chunkang Zhang, Ben He, Xianpei Han, and Le Sun. 2024. Influence of external information on large language models mirrors social cognitive patterns. *IEEE Transactions on Computational Social Systems*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024a. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Xiaoyang Chen, Ben He, Hongyu Lin, Xianpei Han, Tianshu Wang, Boxi Cao, Le Sun, and Yingfei Sun. 2024b. Spiral of silences: How is large language model killing information retrieval?–a case study on open domain question answering. *arXiv preprint arXiv:2404.10496*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729.

Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, Gang Wang, and Jun Xu. 2024. Neural retrievers are biased towards llm-generated content. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 526–537.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Philip Feldman, James R Foulds, and Shimei Pan. 2024. Ragged edges: The double-edged sword of retrieval-augmented chatbots. *arXiv preprint arXiv:2403.01193*.

Chunjing Gan, Dan Yang, Binbin Hu, Hanxiao Zhang, Siyuan Li, Ziqi Liu, Yue Shen, Lin Ju, Zhiqiang Zhang, Jinjie Gu, et al. 2024. Similarity is not all you need: Endowing retrieval augmented generation with multi layered thoughts. *arXiv preprint arXiv:2405.19893*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Yunfan Gao, Yun Xiong, Yijie Zhong, Yuxi Bi, Ming Xue, and Haofen Wang. 2025. Synergizing rag and reasoning: A systematic review. *arXiv preprint arXiv:2504.15909*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance? *arXiv preprint arXiv:2411.10541*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Haoqiang Kang, Juntong Ni, and Huaxiu Yao. 2023. Ever: Mitigating hallucination in large language models through real-time verification and rectification. *arXiv preprint arXiv:2311.09114*.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113.

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*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Cheng Li, Jindong Wang, Kaijie Zhu, Yixuan Zhang, Wenxin Hou, Jianxun Lian, and Xing Xie. 2023. Emotionprompt: Leveraging psychology for large language models enhancement via emotional stimulus. *arXiv e-prints*, pages arXiv–2307.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, et al. 2024a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

Dawei Li, Shu Yang, Zhen Tan, Jae Young Baik, Sunkwon Yun, Joseph Lee, Aaron Chacko, Bojian Hou, Duy Duong-Tran, Ying Ding, et al. 2024b. Dalk: Dynamic co-augmentation of llms and kg to answer alzheimer's disease questions with scientific literature. *arXiv preprint arXiv:2405.04819*.

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

Yi Liu, Lianzhe Huang, Shicheng Li, Sishuo Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Recall: A benchmark for llms robustness against external counterfactual knowledge. *arXiv preprint arXiv:2311.08147*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

Yannic Neuhaus, Maximilian Augustin, Valentyn Boreiko, and Matthias Hein. 2023. Spurious features everywhere-large-scale detection of harmful spurious features in imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20235–20246.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.

Jiejun Tan, Zhicheng Dou, Wen Wang, Mang Wang, Weipeng Chen, and Ji-Rong Wen. 2024a. Htmlrag: Html is better than plain text for modeling retrieved knowledge in rag systems. *arXiv preprint arXiv:2411.02959*.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024b. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957.

Yang Wang, Alberto Garcia Hernandez, Roman Kyslyi, and Nicholas Kersting. 2024. Evaluating quality of answers for retrieval-augmented generation: A strong llm is all you need. *arXiv preprint arXiv:2406.18064*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jinyang Wu, Feihu Che, Chuyuan Zhang, Jianhua Tao, Shuai Zhang, and Pengpeng Shao. 2024a. Pandora's box or aladdin's lamp: A comprehensive analysis revealing the role of rag noise in large language models. *arXiv preprint arXiv:2408.13533*.

Kevin Wu, Eric Wu, and James Zou. 2024b. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*.

Shiping Yang, Renliang Sun, and Xiaojun Wan. 2023a. A new benchmark and reverse validation method for passage-level hallucination detection. In *Findings* *of the Association for Computational Linguistics: EMNLP 2023*, pages 3898–3908.

Shiping Yang, Renliang Sun, and Xiaojun Wan. 2023b. A new dataset and empirical study for sentence simplification in chinese. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8306–8321.

Kui Yu, Xianjie Guo, Lin Liu, Jiuyong Li, Hao Wang, Zhaolong Ling, and Xindong Wu. 2020. Causality-based feature selection: Methods and evaluations. *ACM Computing Surveys (CSUR)*, 53(5):1–36.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Yujia Zhou, Yan Liu, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Zheng Liu, Chaozhuo Li, Zhicheng Dou, Tsung-Yi Ho, and Philip S Yu. 2024. Trustworthiness in retrieval-augmented generation systems: A survey. *arXiv preprint arXiv:2409.10102*.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv e-prints*, pages arXiv–2306.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. Prosa: Assessing and understanding the prompt sensitivity of llms. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1950–1976.

A Related Work

A.1 Robustness Evaluation of Retrieval-Augmented Generation

RAG systems comprise two core components: a retriever and an LLM-based reader. Augmenting LLMs with retrieved external knowledge has been proven to effectively reduce hallucinations (Shuster et al., 2021; Kang et al., 2023). However, the retrieved contexts inevitably contains noise in addition to desirable knowledge, which may mislead LLMs to produce an incorrect response (Bian et al., 2024; Feldman et al., 2024). Previous works have explored automated evaluation frameworks to assess the robustness of RAG systems in various settings. For instance, Chen et al. (2024a) benchmarked four fundamental capabilities required for RAG, including noise robustness, negative rejection, information integration and counterfactual robustness. Some studies have provided a detailed taxonomy of noise documents to further simulate the complexity of real-world scenarios and highlighted the potential positive effects of certain types of noise (Cuconasu et al., 2024; Wu et al., 2024a). There are also some recent works that propose using LLM-as-a-judge (Li et al., 2024a) to evaluate the RAG system (Wang et al., 2024).

While these studies have identified several explicit noises that affect the robustness of RAG systems, they overlook implicit noises. This type of noise, such as phrasing and formatting, is everywhere and unavoidable, as it coexists with the grounding data without altering its semantic information. In this work, we define these semantic-agnostic noises as spurious features and evaluate the robustness of RALMs to such noises.

A.2 Prompt Sensitivity of LLMs

Large Language Models take prompts as inputs and then generate response accordingly. Prompts are instructions provided to an LLM to perform specific tasks automatically and ensure desired qualities in the generated output. However, it is known that current LLMs are sensitive to the features of input prompts (Zhu et al., 2023). This sensitivity poses challenges for researchers attempting to evaluate the model's performance accurately and precisely (Zhuo et al., 2024).

Some existing works have investigated the impact of different prompt techniques on model performance, including chain-of-thought (Wei et al., 2022), in-context learning (Min et al., 2022), and role-play prompting (Kong et al., 2024). Beyond these causal features that significantly influence the meaning of prompts, other works have demonstrated that LLMs are highly sensitive to spurious features (Sclar et al., 2024), e.g, prompt formatting (He et al., 2024), language style (Li et al., 2023), the order of options (Pezeshkpour and Hruschka, 2024).

Currently, there is no statistical or empirical evidence to support the existence of spurious features in RALMs. To address this gap, we extend the definition of spurious features to RAG systems through statistical testing and empirical analysis.

B Preliminary Experiment Results

Using *Contriever-msmarco*, we recall 100 documents from the Wikipedia dump for each query in the NQ-open dataset. After filtering out documents that do not contain golden answers, we select the first-ranked and last-ranked documents based on their oracle scores for each query from the remaining documents, resulting in two sets of 2658 samples each. By comparing the differences in feature distributions between these two sets, we can assess whether RALMs exhibits sensitivity toward semantic-agnostic features. If these two sets do not belong to the same feature distribution, this can be attributed to the inherent bias of LLMs towards semantic-agnostic features. To confirm that this bias is not introduced by the dense retriever in first-stage retrieval, we establish a control group by randomly sampling two documents instead of selecting the first- and last-ranked documents.

To evaluate whether the two distributions are same, we employ the Kolmogorov-Smirnov (K-S) test. The following semantic-agnostic features are measured in the experiments:

- **Flesch Score**: A readability metric designed to evaluate text difficulty. It is calculated based on the average number of syllables per word and the average number of words per sentence. The Flesch score is a number on a scale from 0 to 100, where a higher score indicates that the text is easier to read.

- **Distinct-1**: A metric used to assess the diversity of generated text. It calculates the proportion of unique words (distinct words) to the total number of words in the output. A higher Distinct-1 score indicates that the text contains a greater variety of unique words, implying more diversity in the generated content.

- **Dependency Tree Depth (DTD)**: A syntactic complexity metric calculated by analyzing its dependency tree. Dependency Tree Depth refers to the maximum depth of a sentence's dependency parse tree. A deeper tree suggests more complex sentence structures, while a shallower tree indicates simpler syntactic constructions.

- **Perplexity (PPL)**: A metric used for evaluating language models, measuring how well a probabilistic model predicts a given text. It reflects the uncertainty of a language model when generating sequences of words. Lower PPL values indicate better predictive performance, meaning the model assigns higher probabilities to the actual labels in the sequence.

- **Token Length**: We compute the total number of tokens in a text as an alternative measure of text length, given that the documents in our corpus have been pre-segmented into fixed 100-word chunks. The value is model-specific and depends on the model's vocabulary.

Kolmogorov-Smirnov (K-S) Test The K-S test is a non-parametric statistical test used to compare the distribution of two datasets. It evaluate whether two samples come from the same underlying probability distribution. The null hypothesis of the K-S test is that the two samples are drawn from the same distribution, while the alternative hypothesis is that the two samples are drawn from different distributions. There are two key values provided by K-S test: the K-S Statistic quantifies the largest difference between the two sample distributions, and the p-value assess the statistical significance of that difference. If the p-value is lower than a chosen significance level (0.05 in our experiments), we reject the null hypothesis, concluding that the two distributions are significantly different. Otherwise, we fail to reject the null hypothesis, suggesting that there is no significant difference between the two distributions.

The K-S statistic and P-value are presented in Table 4 and Table 5. Furthermore, we visualize the feature distributions for both the experimental and control groups in Figure 6. For all tested features in the experimental group, the K-S test rejects the null hypothesis, concluding that the distribution of the two sets are significantly different. In contrast, for the control group, the K-S test fails to reject the null hypothesis. The results for *Llama-3.1-8B-Instruct* are also provided in Figure 7. According to these results, we can conclude that RALMs exhibit bias toward spurious features in documents.

	Experimental Group		Control Group	
	K-S statistic	P-value	K-S statistic	P-value
Flesch score	0.0677	$1.01 \times 10^{-5***}$	0.0301	0.1799
Distinct-1	0.0756	$4.95 \times 10^{-7***}$	0.0203	0.6431
DTD	0.0636	$4.29 \times 10^{-5***}$	0.0124	0.9866
PPL	0.0722	$1.88 \times 10^{-6***}$	0.0162	0.8776
Token Length	0.1708	$2.91 \times 10^{-34***}$	0.0256	0.3493

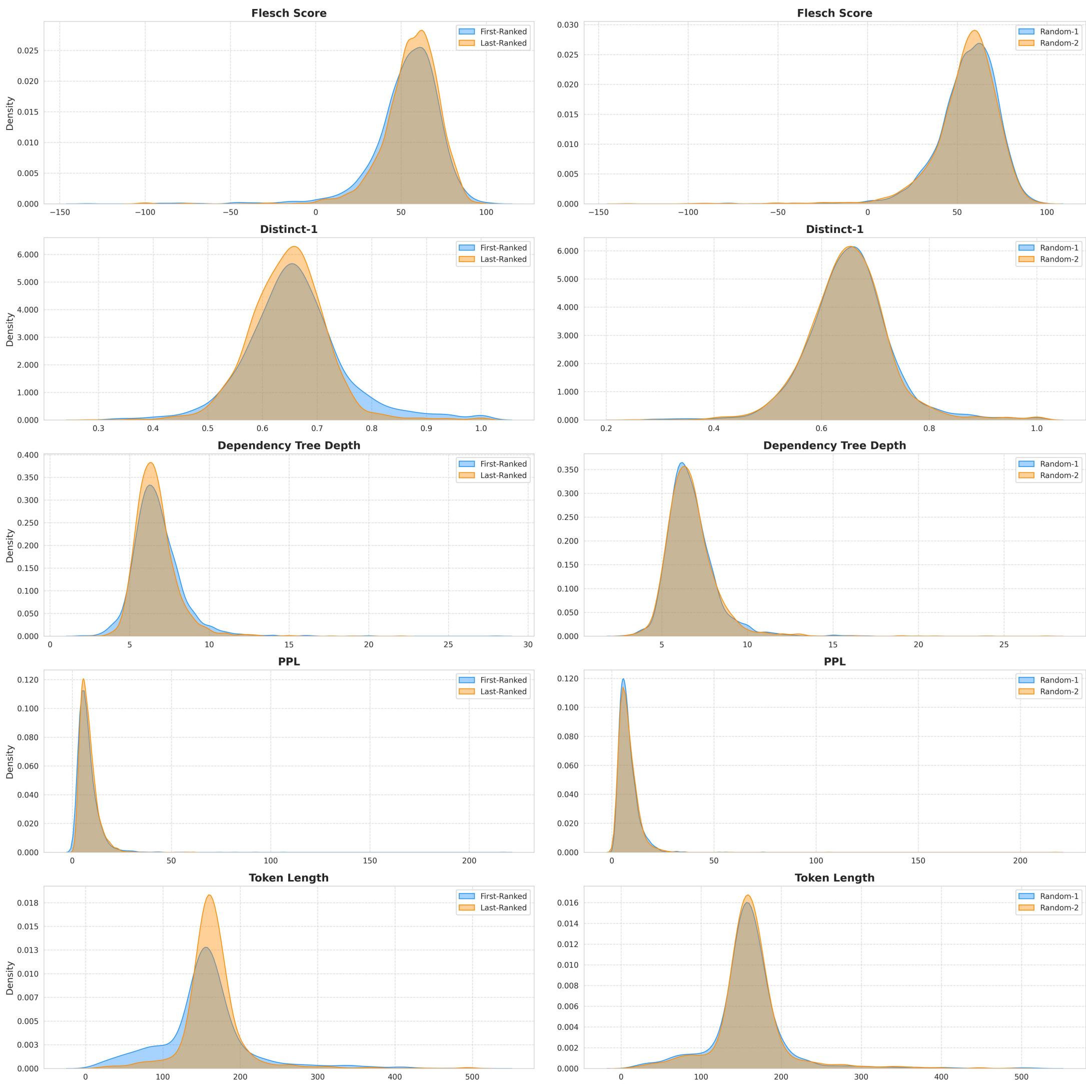
Table 4: K-S test results for *Mistral-7B-Instruct-v0.3* as the oracle retriever.

C Implementation Details for Injecting Spurious Features

We provide detailed prompts for LLM-based perturbations in Figure 8. For rule-based perturbations, placeholder template is presented in Figure 9.

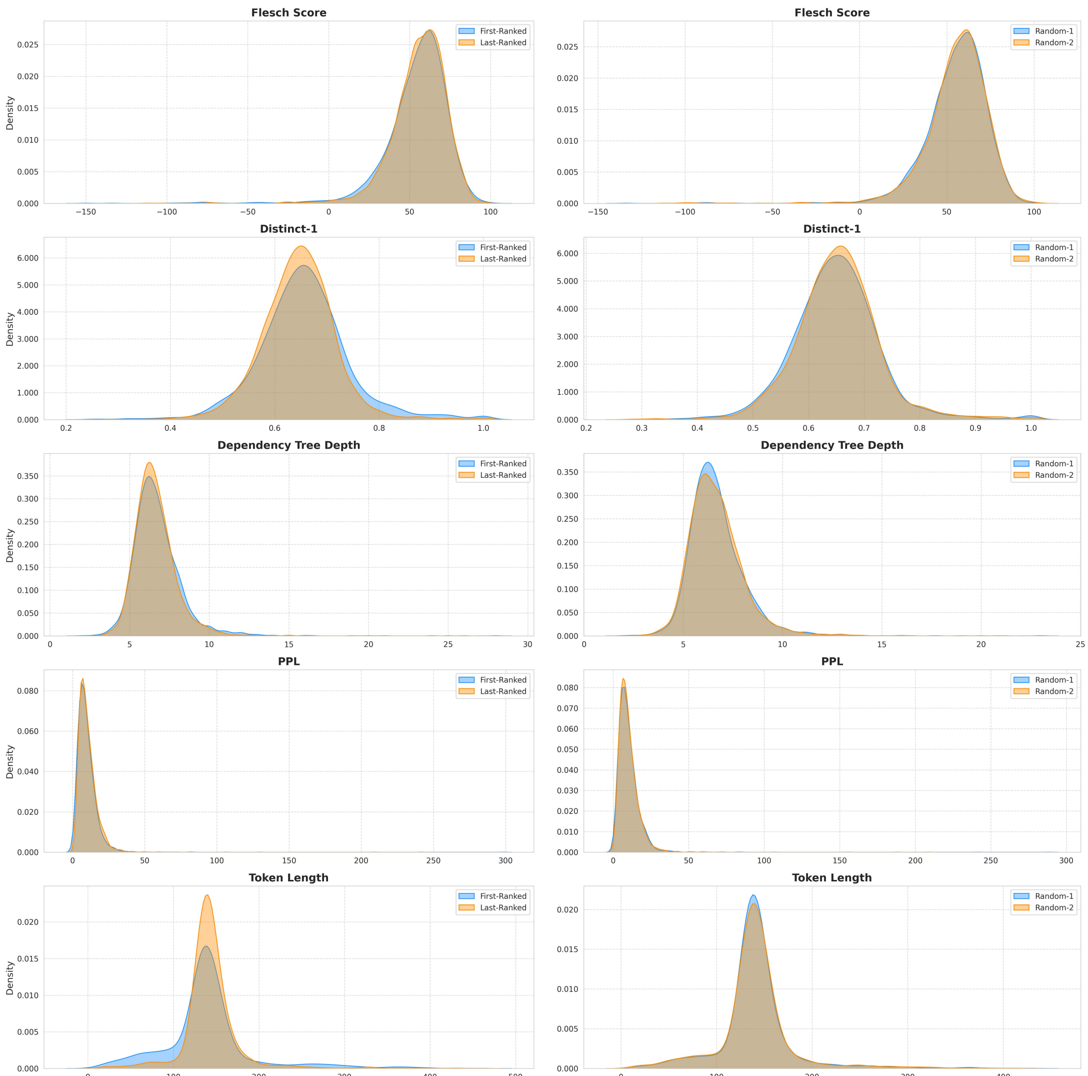
	Experimental Group		Control Group	
	K-S statistic	P-value	K-S statistic	P-value
Flesch score	0.0305	0.1694	0.0173	0.8210
Distinct-1	0.0798	$8.94 \times 10^{-8***}$	0.0327	0.1159
DTD	0.0474	0.0051^{**}	0.0203	0.6431
PPL	0.0538	0.0009^{***}	0.0181	0.7791
Token Length	0.1275	$2.99 \times 10^{-19***}$	0.0188	0.7349

Table 5: K-S test results for *Llama-3.1-8B-Instruct* as the oracle retriever.



(a) Feature distributions of the experimental group (b) Feature distribution of the control group

Figure 6: Visualization of feature distributions for *Mistral-7B-Instruct-v0.3*



(a) Feature distributions of the experimental group

(b) Feature distribution of the control group

Figure 7: Visualization of feature distributions for *Llama-3.1-8B-Instruct*

Style Perturbations

[Simple]
Please simplify the following text while preserving its original meaning. Use shorter sentences, basic vocabulary, and clear language. Avoid complex structures, technical terms, or ambiguous expressions.
Here is the passage to simplify:{Document}
[Complex]
Please complexify the following text while preserving its original meaning. Use longer sentences, intricate sentence structures, and advanced vocabulary. Avoid contractions, informal language, and colloquial expressions, ensuring the text maintains a professional and authoritative tone throughout.
Here is the passage to complexify:{Document}

Source Perturbations

Please rewrite the following passage. Ensure that the overall meaning, tone, and important details remain intact. Avoid any significant shifts in style or focus. The aim is to create a fresh version while faithfully conveying the original content.
Here is the passage to paraphrase:{Document}

Logic Perturbations

[LLM-Ranked]
Rearrange the following list of sentences in your preferred logical order and provide only the indices of the sentences. Please do not include any explanations.
Example:{Example}
Sentences List:{Sentences List}
The length of the Sentences List is {Length of Sentences List}. Therefore, the indices must contain {Length of Sentences List} elements, and the index values cannot exceed {Length of Sentences List - 1}.
[Reverse] [Pyhton Code]
[Random] [Python Code]

Figure 8: Prompt templates for LLM-based perturbations.

Format Perturbations

[JSON]

```
{
    "title": "{Title}",
    "text": "{Document}"
}
```

[HTML]

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    {Title}
</head>
<body> {Document} </body>
</html>
```

[YAML]

```
Title: {Title}
Text: {Document}
```

[Markdown]

```
# {Title}
{Document}
```

Metadata Perturbations

[Timestamp]

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name='timestamp' content='{timestamp}'>
    {Title}
</head>
<body> {Document} </body>
</html>
```

[Datasource]

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name='datasource' content='{datasource}'>
    {Title}
</head>
<body> {Document} </body>
</html>
```

Figure 9: Placeholder templates for rule-Based perturbations.

D Implementation Details for Preserving Causal Features

We employ a bidirectional entailment algorithm to ensure the semantic equivalence before and after introducing spurious features. The prompts for its core component, NLI model, are shown in Figure 10. Furthermore, we apply a simple string-matching strategy to preserve ground truths. Specifically, for *Golden* documents that originally contained the correct answers, we keep them only if they preserve the ground truths after perturbation. For *Noise* documents that initially lack the correct answers, we discard them if they unexpectedly acquire ground truths due to perturbations.

> Consider the two passages below.
> Premise: {raw text}
> Hypothesis: {perturbated text}
> Does the premise semantically entail the hypothesis? Answer with 'entailment' if they are paraphrases,'contradiction' if they have opposing meanings, or 'neutral' if they are neither.
> Response:

Figure 10: Prompts for LLM-based NLI system.

E Statistics of the Synthetic Dataset

We present the dataset statistics for evaluating *Llama-3.1-8B-Instruct* in Table 6.

	K-G	**K-N**	**U-G**	**U-N**	**Total**
Style	7321	28975	3038	39869	79203
Source	8768	30145	3709	41391	84013
Logic	9229	33294	4082	44233	90838
Format	10481	35616	4697	47920	98714
Meta	10563	35451	4796	47987	98797

Table 6: Distribution of the *SURE_Wiki* dataset for *Llama-3.1-8B-Instruct*.

F Experimental Setup Details

Prompts The instruction I in the RAG prompt $P = (I, G, Q)$, shown in Figure 11, is derived from Cuconasu et al. (2024), with slight modifications to better adapt to our setting.

Implementation Details We follow the typical "retrieve-read" setting of RAG paradigm. For the retrieval module, we use *Contriever-msmarco*[7], a BERT-based dense retriever, as the default retriever. It is finetuned on the MS MARCO dataset (Bajaj et al., 2016) after unsupervised pretraining via contrastive learning (Izacard et al., 2021). To optimize the efficiency of vector similarity searches, we employ the Faiss library (Douze et al., 2024). For the read module, we deploy LLMs on NVIDIA A100 GPUs and accelerate inference with vllm[8]. We set the temperature to 0.1 to ensure stable outputs and strong reproducibility.

> You are given a question and you MUST respond by EXTRACTING the answer (max 5 tokens) from the provided document. If the document does not contain the answer, respond with NO-RES.

Figure 11: Instruction I used for the QA task.

Llama-3.1-8B-Instruct																	
Taxonomy	**Perturbations**	**Known-Golden**					**Known-Noise**					**Unknown-Golden**					**U-N**
		LR	RR	WR	Org	Acc	LR	RR	WR	Org	Acc	LR	RR	WR	Org	Acc	RR
Style	Simple	7.79	83.04	**9.18**	66.03	67.42	1.70	95.80	**2.50**	4.12	4.92	8.43	82.88	**8.69**	51.42	51.68	99.45
	Complex	6.00	85.60	**8.40**		68.43	1.91	96.59	1.50		3.71	6.71	84.86	**8.43**		53.13	99.57
Source	LLM-Generated	5.89	86.43	**7.69**	65.62	67.43	1.43	96.83	**1.74**	4.13	4.45	6.20	85.71	**8.09**	49.15	51.04	99.56
	Self-Generated	6.55	85.01	**8.44**		67.52	1.55	96.37	**2.09**		4.67	6.52	86.36	**7.12**		49.74	99.57
Logic	Reverse	5.06	90.82	4.12	62.95	62.01	1.13	97.82	1.06	4.43	4.36	5.73	89.71	4.56	45.84	44.66	99.67
	Random	3.91	93.16	2.93		61.97	0.86	98.31	0.83		4.40	4.21	91.67	4.12		45.74	99.72
	LLM-Ranked	3.24	93.93	2.83		62.54	0.82	98.43	0.74		4.36	3.58	93.36	3.06		45.32	99.76
Format	JSON	7.01	88.25	4.74	63.91	61.64	1.70	97.25	1.05	3.87	3.21	5.92	89.63	4.45	49.35	47.88	99.61
	HTML	11.85	84.46	3.69		55.75	2.70	96.90	0.40		1.56	9.33	86.78	3.90		43.92	99.61
	YAML	5.26	89.94	4.80		63.45	1.26	97.41	1.33		3.94	4.79	90.80	4.41		48.97	99.67
	Markdown	2.32	92.23	**5.45**		67.04	0.60	96.89	**2.51**		5.77	2.34	93.46	**4.19**		51.20	99.61
Metadata	Timestamp (pre)	2.08	95.81	**2.11**	55.77	55.80	0.28	99.42	**0.29**	1.58	1.59	2.54	95.56	1.90	43.31	42.66	99.95
	Timestamp (post)	2.04	95.86	**2.10**		55.84	0.25	99.43	**0.32**		1.64	2.81	95.56	1.63		42.12	99.95
	Datasource (wiki)	2.11	93.45	**4.44**		58.10	0.23	98.96	**0.81**		2.17	3.25	92.47	**4.27**		44.33	99.86
	Datasource (twitter)	2.27	94.11	**3.62**		57.11	0.31	99.25	**0.43**		1.70	2.77	93.97	**3.25**		43.79	99.91

Table 7: Robustness evaluation results of *Llama-3.1-8B-Instruct* on the synthetic dataset. We use **Bold** to mark the WR values that are higher than the LR, suggesting that the perturbation is beneficial.

[7] https://huggingface.co/facebook/contriever-msmarco
[8] https://github.com/vllm-project/vllm