Don't Corrupt the Fact: A Trustworthy RAG Watermarking Framework based on Dual Factual Shield

Hao Huang[1,2] , Jiatang Luo[3,4] , Ruihua Zhou[1,2] , Yunpeng Li[2]* , Yuling Liu[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]School of Advanced Interdisciplinary Sciences,
University of Chinese Academy of Sciences, Beijing, China
[4]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{huanghao, zhouruihua, liyunpeng, liuyuling}@iie.ac.cn, luojiatang@ict.ac.cn

Abstract

While RAG systems are designed to enhance factual fidelity by grounding LLMs in provided sources, the application of current watermarking techniques creates a conflict. These methods, being inherently fact-agnostic, force the model to deviate from the very source documents it is supposed to follow. This leads to "faithfulness hallucinations", which refer to a critical flaw where the generated output contradicts its own grounding context. Consequently, these watermarks undermine the core value of RAG, rendering even the most secure schemes untrustworthy for high-stakes applications. To resolve this RAG-specific conflict, we introduce the Dual Factual Shield (DFS), a three-stage post-hoc pipeline for factuality-preserving watermarking in RAG. It adopts a defense-in-depth design that combines a source-anchored algorithmic safeguard for protecting critical tokens from retrieved context with prompt-based semantic guidance to mitigate factual corruption. Experiments show that our framework drastically reduces the Knowledge Corruption Rate (KCR), a new metric we introduce to quantify factual fidelity, while maintaining strong security and robustness, paving the way for responsible deployment of traceable AI in knowledge-critical domains.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in generating human-like text (Bi et al., 2024; Yang et al., 2025; Dubey et al., 2024), powering a new generation of applications from chatbots to sophisticated information retrieval systems (Zhao et al., 2025). Among these, RAG has emerged as a widely adopted paradigm, enhancing the factuality and relevance of LLM outputs by grounding them in external knowledge bases (Borgeaud et al., 2022; Lewis et al., 2020). However, the ease of generating high-quality text also introduces significant risks, such as the spread of misinformation and copyright violations. To mitigate these risks, watermarking (Kirchenbauer et al., 2023; Aaronson and Kirchner, 2022) has been proposed as a promising mechanism for tracing the provenance of machine-generated content.

Figure 1: An example of a "faithfulness hallucination".

Recent advancements in watermarking have produced increasingly sophisticated algorithms. Significant progress has been made on robustness against paraphrasing and removal attacks (Kuditipudi et al., 2023; Christ et al., 2023). More recently, to combat malicious "spoofing" attacks, where an adversary alters a watermarked text's meaning to frame an RAG provider, *semantic-aware* watermarks have been developed (Liu et al., 2024; An et al., 2025). These methods, often using techniques like contrastive learning to train a semantic-aware model, can successfully invalidate a watermark when the text's core meaning is dis-

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 44816–44826

torted, thus providing a crucial layer of security.

However, despite their sophistication, even these SOTA secure watermarking methods suffer from a fundamental, unaddressed flaw in the context of RAG: a lack of knowledge loyalty. While designed to protect semantic integrity, they remain inherently fact-agnostic (Lee et al., 2024; Zhao et al., 2024; Wu et al., 2024; An et al., 2025). This leads to the critical failure mode we term a "faithfulness hallucination," vividly illustrated in Figure 1. The "Inhonest" response in the figure contains multiple, disastrous factual errors directly contradicted by the source text. This happens because existing watermarks are fact-agnostic. Their underlying mechanism indiscriminately partitions the vocabulary into "green" and "red" lists. In this random assignment, a critical token like the year "*2023*" has no special protection and can be allocated to the "red" list. As the model is then steered to sample from the green list, it is effectively penalized for selecting the correct fact, often leading it to output a plausible but incorrect alternative. This inherent risk of knowledge distortion undermines the core value proposition of RAG and erodes user trust (Ji et al., 2023), making even the most secure watermarks unsuitable for deployment in high-stakes settings.

To resolve this RAG-specific conflict between traceability and faithfulness, we introduce the Dual Factual Shield (DFS) framework, a novel architecture built upon the core principle of knowledge loyalty. The DFS framework is designed as a defense-in-depth system, combining "soft" semantic guidance to steer the LLM with a "hard" algorithmic safeguard that significantly reduces the risk of altering critical facts. The goal of this framework is to prove a core principle: that with the right architecture, security and trustworthiness can, and must, coexist. We validate this principle by instantiating DFS with a spoofing-robust semantic backbone, demonstrating that the resulting system fundamentally enhances trustworthiness. In our experiments, DFS is evaluated on WikiEval, a standard RAG benchmark of question–context–answer triples, and on Wiki-Big, a larger-scale dataset we construct to stress-test factual preservation under diverse, entity- and number-heavy contexts.

Our main contributions are as follows:

- We are the first to identify and systematically address the problem of watermark-induced knowledge distortion in RAG systems, a paradoxical flaw where traceability mechanisms undermine factual faithfulness.
- We design and propose the novel Dual Factual Shield framework, a hybrid architecture combining semantic guidance with a source-anchored algorithmic safeguard to substantially improve RAG-specific factual integrity.
- We introduce the Knowledge Corruption Rate (KCR) as a new, essential metric for evaluating the factual fidelity of watermarked text against its source context.
- An experimental demonstration across WikiEval and Wiki-Big that DFS maintains strong detectability and improves spoofing security while substantially improving factuality (e.g., on WikiEval: KCR drops from 14.67% to 5.56%, and Faithfulness rises from 74.34% to 86.99%).

2 Related Work

2.1 General LLM Watermarking

LLM watermarking has emerged as a critical technique for addressing the authenticity and accountability challenges posed by AI-generated content, enabling the detection and verification of machine-generated text to combat misinformation and ensure content traceability. Existing watermarking approaches can be broadly divided into two categories: integrating with LLM generation and post-processing after LLM generation. The first category injects watermarks during the text generation process, specifically at three sequential stages: LLM training (Gu et al., 2024; Xu et al., 2024a,b), logits generation (Kirchenbauer et al., 2023; Lee et al., 2024; Hu et al., 2024; Zhao et al., 2024), and token sampling (Hou et al., 2024). The logits generation stage has attracted the most research attention due to its effectiveness and practicality. KGW (Kirchenbauer et al., 2023) uses a hash-based mechanism to partition vocabulary into red and green lists and employs z-score analysis for watermark detection. The second category applies watermarking in a two-pass, post-hoc manner: it first produces an unwatermarked draft and then rewrites/edits it in a second pass to embed the watermark, leveraging access to the full draft for more global control. REMARK-LLM (Zhang et al., 2024) leverages a representation space to combine the generated text

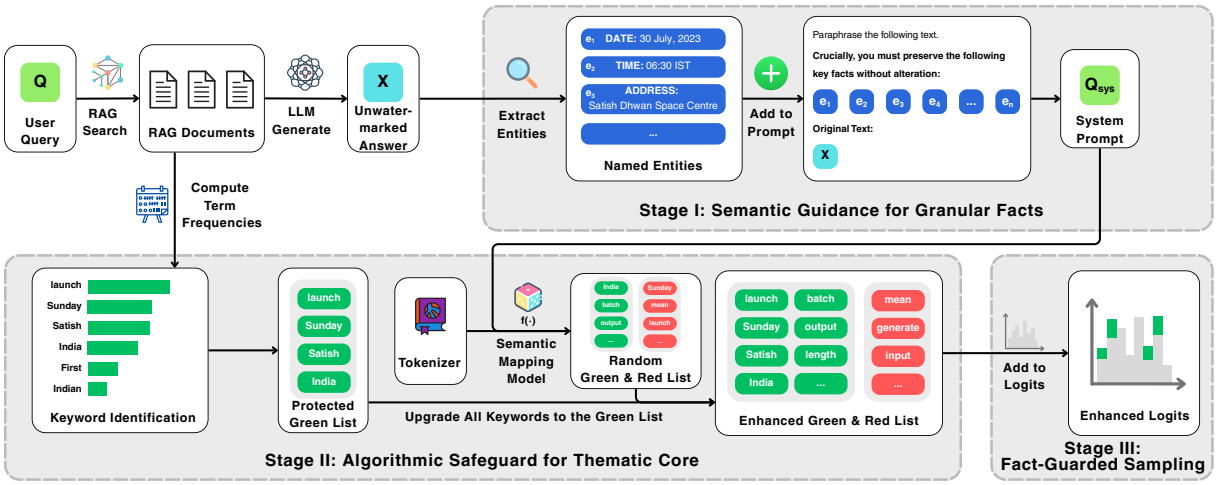

Figure 2: Overview of the DFS algorithm. Stage I provides semantic guidance for granular facts; Stage II constructs the fact-guarded greenlist using a contrastive semantic encoder combined with source-grounded constraints; Stage III performs fact-guarded sampling to produce the final watermarked output.

with its signature and induces a watermarked token distribution for subsequent generation. Recent work further strengthens robustness against spoofing attacks through contrastive representation learning (An et al., 2025). However, these methods are predominantly designed for general text generation scenarios and fail to consider the unique characteristics of RAG systems, including their strong dependence on retrieved contextual information and high requirements for factual accuracy, necessitating the development of specialized watermarking approaches for RAG environments.

2.2 The Unaddressed Challenge: Factual Corruption in Watermarking

RAG systems mitigate hallucinations by grounding generation in retrieved documents (Fan et al., 2024). However, existing watermarking methods rely on statistical randomness to embed signals, which may inadvertently alter critical factual information such as years, numbers, or names. Prior work has focused primarily on detectability and robustness (Yoo et al., 2023; Kirchenbauer et al., 2023; Lee et al., 2024; Zhao et al., 2024), largely overlooking factual fidelity as a design principle. Our work addresses this gap by proposing a fact-aware mechanism that actively protects key factual tokens during watermark embedding.

3 Methodology

We propose the DFS algorithm, a three-stage post-hoc pipeline that rewrites an unwatermarked draft answer X into a watermarked answer X_w while preserving source-grounded facts. The pipeline is designed to achieve both spoofing-robust traceability and knowledge loyalty—the property of preserving the factual integrity of the generated text—and proceeds as follows: **Stage I** provides semantic guidance to steer the LLM toward factual preservation; **Stage II** constructs a fact-guarded greenlist by combining a spoofing-robust semantic encoder with source-grounded constraints; and **Stage III** applies logit biasing with this protected greenlist to produce the final watermarked output.

3.1 Stage I: Fact-Preserving Generation (Soft Constraint)

The first stage provides fact-centric guidance to steer the LLM toward factual preservation during generation. A key design challenge is balancing the scope of factual protection with the statistical integrity of the watermark—an overly aggressive protection strategy can dilute the watermark signal. Therefore, this stage acts as a "soft" context-aware guide rather than a rigid constraint.

Fact Extraction. We employ the spaCy toolkit (Honnibal and Montani, 2017) as a lightweight entity recognition model to automatically extract named entities (e.g., persons, organizations, locations) and use rule-based pattern matching to identify numerical entities (e.g., dates, percentages, monetary values). Let the set of these extracted factual phrases be denoted by $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$.

Instruction Augmentation. We then augment the paraphrasing prompt Q_{sys}, given to the LLM

with an explicit instruction to retain these facts. This augmented prompt acts as a "soft constraint," priming the LLM to focus on factual preservation and reducing the likelihood of accidental knowledge corruption during text generation.

Algorithm 1 Fact-Preserving Semantic Guidance

Require: draft answer X, NER model M_{NER}, numeric extractor Φ_{NUM}
Ensure: Augmented prompt Q_{sys}
Initialize: $\mathcal{E} \leftarrow \emptyset$; $\mathcal{E}_{\text{str}} \leftarrow \emptyset$; $\mathcal{E}_{\text{num}} \leftarrow \emptyset$
▷ *Step 1: Extract Facts*
$doc \leftarrow M_{\text{NER}}(X)$
for all $(e, k) \in \text{ENTITIES}(doc)$ **do**
  $\mathcal{E} \leftarrow \mathcal{E} \cup \{e\}$
  $\mathcal{E}_{\text{str}}[k] \leftarrow \mathcal{E}_{\text{str}}[k] \cup \{e\}$
end for
$\mathcal{E}_{\text{num}} \leftarrow \Phi_{\text{NUM}}(X)$
$\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{E}_{\text{num}}$
▷ *Step 2: Format System Prompt*
$L \leftarrow \text{SERIALIZEFACTS}(\mathcal{E}_{\text{str}}, \mathcal{E}_{\text{num}})$
$Q_{\text{sys}} \leftarrow \text{AUGMENTPROMPT}(X, L)$
return Q_{sys}

3.2 Stage II: Semantic Mapping & Fact-Guarded Greenlist Construction

While prompt engineering provides effective guidance, it is not foolproof. Stage II constructs the protected greenlist G_{final} through two sub-steps: first computing an initial spoofing-robust greenlist using a semantic mapping encoder, then enforcing source-grounded factual constraints to provide a strong protective bias at the token level. Note that this mechanism significantly reduces the risk of factual token replacement but does not constitute a complete factual consistency guarantee, as its effectiveness depends on the accuracy of the upstream entity extraction.

3.2.1 Spoofing-Robust Semantic Mapping

We instantiate Stage II with a spoofing-robust semantic mapping watermark backbone following An et al. (2025). This choice is orthogonal to DFS; in experiments we report it as the backbone. This encoder, trained via contrastive learning, produces a semantic embedding that is *insensitive* to meaning-preserving modifications (ensuring robustness) while being *sensitive* to meaning-distorting modifications (ensuring security). Given the draft answer X, the encoder $f(\cdot)$ generates a vocabulary-sized vector used to partition the vocabulary V into an initial greenlist G_{base} and a red list R. The LLM's output logits are then perturbed to increase the probability of sampling tokens from G_{base}.

While effective at securing against spoofing, this semantic mapping is fundamentally fact-agnostic—the generation of G_{base} does not account for the preservation of critical factual tokens. Our source-grounded constraint mechanism directly addresses this limitation.

Algorithm 2 Algorithmic Safeguard

Require: Document set D, NER model M_{NER}, tokenizer τ, Vocabulary V
Ensure: Green-token mask $g \in \{0,1\}^{|V|}$
$c \leftarrow \mathbf{0}$ ▷ count vector over V
for all $d \in D$ **do**
  $\mathcal{S} \leftarrow M_{\text{NER}}(d)$
  for all $s \in \mathcal{S}$ **do**
    for all $t \in \tau(s)$ **do**
      $c[t] \leftarrow c[t] + 1$
    end for
  end for
end for
$E \leftarrow \{t \in V \mid c[t] > 0\}$
$g_{\text{fact}} \leftarrow \mathbf{0}$
for all $t \in E$ **do**
  $g_{\text{fact}}[t] \leftarrow 1$
end for
return g_{fact}

3.2.2 Source-Grounded Factual Constraints

A core feature of RAG is its reliance on a source knowledge base. We leverage this by identifying and protecting all salient factual terms from the original RAG context documents. Our strategy focuses exclusively on Named Entity Recognition, employing spaCy to extract a comprehensive set of entities, including persons, locations, and organizations. Critically, we include every unique entity token that appears at least once in the retrieved evidence, while stopwords and function words (e.g., "the", "a", "of") are excluded by construction since they are not recognized as named entities. This approach ensures that even facts mentioned only once, which are common in diverse RAG sources, are treated as essential. This entity-derived token set is then formally denoted as S_{fact}.

Fact-Guarded Greenlist Construction. This is the core of our source-grounded constraint mechanism. Using the initial greenlist G_{base} from the semantic mapping encoder, we enforce a "Do-Not-Alter" policy by creating our protected greenlist, G_{final}, as the union of the encoder's output and our source-grounded keywords:

$$G_{\text{final}} = G_{\text{base}} \cup S_{\text{fact}}. \tag{1}$$

By forcing all tokens in S_{fact} into the greenlist, we significantly increase their sampling probability and reduce the likelihood of their replacement

by red-list tokens. A key insight is that because the tokenizer decomposes entity strings into a constrained set of tokens, this policy does not lead to an overly bloated greenlist that could dilute the watermark signal. This provides a robust protective bias that strongly favors the preservation of core terminology, even if the LLM deviates from the prompt guidance.

3.3 Stage III: Fact-Guarded Sampling

Using the protected greenlist G_{final} constructed in Stage II, we apply logit biasing to the LLM's output distribution, increasing the probability of sampling tokens from G_{final}. This produces the final watermarked text X_w. Detection follows the standard statistical detection framework of the underlying semantic encoder, where the presence of the watermark is verified by measuring the proportion of greenlist tokens in the text.

The complete pipeline of our DFS algorithm is illustrated in Figure 2. By intrinsically coupling source-grounded factual protection with spoofing-robust semantic mapping, our algorithm fundamentally resolves the knowledge distortion problem while maintaining high security. This yields a robust, secure, and demonstrably trustworthy watermarking solution.

4 Experiments

4.1 Experimental Setup

Datasets and Backbone LLM. Our evaluation targets retrieval-grounded settings where answers must remain fully supported by the provided context, and where even minor perturbations to entities or numbers constitute measurable factual corruption. We utilize the WikiEval dataset (Es et al., 2024), which consists of question-context-answer triples constructed from English Wikipedia. We additionally evaluate on Wiki-Big, a multi-domain (including professional domains such as finance and healthcare) Wikipedia-based benchmark we construct to cover diverse industry entities and number-heavy factual details. Construction details and dataset statistics are provided in Appendix A.2. In our main experiments, we use 50 queries from WikiEval and 300 queries from Wiki-Big. For our text generation and paraphrasing tasks, we primarily utilize Llama-3.1-8B as our backbone LLM. Additionally, we employed Qwen2.5-0.5B and Qwen2.5-7B in select experiments for comparative analysis.

Baseline To demonstrate the effectiveness and generality of the DFS algorithm, we compare it against a diverse set of representative watermarking paradigms: KGW (Kirchenbauer et al., 2023), Unigram (Zhao et al., 2024), SWEET (Lee et al., 2024), Unbiased (Hu et al., 2024), DiP (Wu et al., 2024), and Contrastive WM (An et al., 2025). Detailed descriptions for each of these methods are provided in Appendix B.

Implementation Details. Our DFS algorithm is a three-stage post-hoc watermarking pipeline; Stages I–II are implemented as lightweight preprocessing over prompts and greenlist construction, followed by Stage III fact-guarded sampling. We utilize a single NER model—spaCy's `en_core_web_sm`—for both Stage I and Stage II. In Stage I, this model identifies entities within the initial LLM-generated answer. In Stage II, it processes the retrieved source documents to define key terms for protection, such as persons, organizations, and other semantic entities. In our experimental setup, all baseline methods were configured using the default hyperparameters(e.g., γ, δ) specified in the MarkLLM framework. All experiments are conducted on a server equipped with two NVIDIA RTX 4090 GPUs and 128GB of RAM, operating on Ubuntu 22.04.5 LTS.

4.2 Evaluation Metrics

Faithfulness The Faithfulness metric quantifies how factually grounded a generated answer is in its provided context, aiming to penalize model hallucinations. Following the Ragas (Es et al., 2024) methodology, this score is calculated by using a LLM to first decompose the answer into a set of atomic statements, and then verifying how many of these statements are supported by the context.

The final score is the ratio of supported statements to the total, as shown in Equation 2. A higher score, ranging from 0 to 1, indicates greater factual accuracy and trustworthiness.

$$\text{Faithfulness} = \frac{VS}{TS}. \tag{2}$$

where:

- VS **(Verified Statements)**: The number of statements verified as "Yes".
- TS **(Total Statements)**: The total number of extracted statements.

Dataset	Metric	No	Llama-3.1-8B							Qwen-2.5-7B						
			KGW	UNI.	SWEET	Unbias	DiP	C. WM	Ours	KGW	UNI.	SWEET	Unbias	DiP	C. WM	Ours
WikiEval	AUC	–	**100**	98.12	**100**	99.40	99.48	98.60	90.68	**99.52**	96.64	98.84	95.28	94.84	98.26	94.62
	KCR	1.52	15.39	20.74	22.80	16.20	14.45	14.67	**5.56**	18.38	18.85	29.54	15.18	24.21	10.67	**8.53**
	Faithfulness	92.76	78.53	74.19	68.67	71.66	71.21	74.34	**86.99**	67.60	66.66	62.19	64.35	61.51	78.66	**83.26**
	ROUGE-1	62.95	42.11	41.22	41.39	43.18	44.35	39.91	**46.25**	39.12	38.74	39.11	39.97	39.99	37.85	**48.49**
	ROUGE-L	55.12	28.70	27.98	27.72	29.14	28.88	27.41	**33.11**	26.12	26.40	26.10	27.18	27.12	28.89	**38.17**
	PPL	11.15	10.50	11.74	12.27	9.88	**9.72**	21.57	16.85	12.79	11.94	14.48	12.01	**11.11**	28.76	17.39
Wiki-Big	AUC	–	99.61	99.85	**100**	99.55	99.62	96.15	94.70	95.30	93.83	97.80	91.18	91.73	96.71	95.88
	KCR	0.28	12.89	16.80	17.39	14.17	16.05	20.28	**8.78**	43.16	44.00	45.56	40.36	40.05	24.00	**14.50**
	Faithfulness	95.52	66.98	63.91	62.86	65.44	65.04	67.39	**76.30**	45.57	45.36	46.08	47.72	46.86	66.65	**75.53**
	ROUGE-1	54.51	16.05	14.79	12.67	15.28	15.31	33.45	**36.53**	18.97	18.20	18.21	18.51	17.90	45.31	**49.20**
	ROUGE-L	50.89	13.77	12.57	10.71	13.16	13.24	27.14	**28.95**	16.84	16.00	16.06	16.22	15.81	40.68	**44.79**
	PPL	14.85	8.11	10.25	11.16	7.67	**7.50**	14.91	17.25	11.49	**11.10**	15.04	11.74	11.63	12.98	14.14

Table 1: Results on **WikiEval** and **Wiki-Big** in a single table. Lower is better for KCR/PPL; higher is better for others. (The column "No" denotes the unwatermarked generation by Llama-3.1-8B. "UNI." is UNIGRAM, "C. WM" is Contrastive WM. Results are reported from a single run unless otherwise specified.)

Watermark Performance. We use the Area Under the ROC Curve (AUC) to evaluate watermark performance on Detectability and Security. Detectability measures the ability to differentiate watermarked from unwatermarked text, where a higher AUC is better. For Security, we evaluate the system's capacity to identify spoofing attacks by classifying between original, valid watermarked texts (positive class) and their maliciously altered versions (negative class). Therefore, a higher Security AUC also signifies a better outcome, indicating a stronger ability to detect semantic corruption and protect against content tampering.

Factual Fidelity. As argued in the introduction, standard metrics are insufficient. To directly quantify factual integrity, we propose and report the Knowledge Corruption Rate (KCR). KCR is the percentage of factual tokens (identified entities and numbers from the standard answer) that are altered or absent in the final watermarked text. A lower KCR is better, with a score of 0 indicating perfect factual preservation. The KCR is calculated as:

$$\text{KCR} = \frac{TF - PF}{TF}. \tag{3}$$

where:

- TF **(Total Facts)**: The number of factual entities extracted from the standard answer.
- PF **(Preserved Facts)**: The number of factual entities that remain consistent between the standard answer and the response texts.

Generation Quality (ROUGE, PPL) To quantify the surface-level quality of watermarked answers, we additionally report ROUGE and perplexity. ROUGE-1 and ROUGE-L are computed as F1 scores of unigram and longest-common-subsequence overlap, respectively, between the generated answer and the reference answer on the evaluation set. Higher ROUGE indicates that the watermarked answer better preserves the lexical content and structure of the original answer. For fluency, we report perplexity (PPL) of the generated answers. Concretely, we score each answer under the corresponding unwatermarked backbone LLM, average the token-level negative log-likelihood, and exponentiate it to obtain PPL. Lower PPL reflects more fluent and natural text.

4.3 Main Results

Our primary findings in Table 1 reveal that unlike baseline methods, which corrupt factual knowledge in a RAG context, our DFS method improves factual fidelity while preserving high performance in security and robustness.

4.3.1 Drastic Reduction in Factual Corruption

Across WikiEval, Wiki-Big, and two backbone LLMs, DFS consistently lowers KCR and improves Faithfulness while preserving strong AUC (Table 1). Most notably, on WikiEval with Llama-3.1-8B, DFS reduces KCR from 14.67% to 5.56% and lifts Faithfulness from 74.34% to 86.99%—a substantial gain in factual preservation. A similar trend holds for Wiki-Big, confirming that our dual-shield mechanism remains highly effective even on larger-scale, knowledge-intensive tasks.

4.3.2 Minimal and Acceptable Performance Trade-offs.

Because S_{fact} is added to the greenlist, unwatermarked outputs overlap more with protected tokens,

Text Version	Generated Answer	Faithfulness	WM
Correct Answer	PharmaCann was founded in **2014**. Its headquarters is located in the state of **Illinois**.	-	-
Our Method	PharmaCann was launched in the year of **2014**, with the company's operational base situated in **Illinois**.	0.955	Yes
Corrupted/Spoofed	PharmaCann was **doomed** to launch in the year of **2016**, with the company's operational base situated in **Indiana**.	0.542	No

Table 2: Qualitative example of a watermarked response. **Underlined text** highlights several errors introduced by the spoofing attacks. Our method preserves factual correctness.

slightly increasing FPR and lowering AUC from 98.60% to 90.68%. However, AUC remains above 90%, an acceptable trade-off for factual fidelity. Similarly, the modest PPL increase reflects tension between strict fact adherence and fluency, yet the output remains natural and coherent.

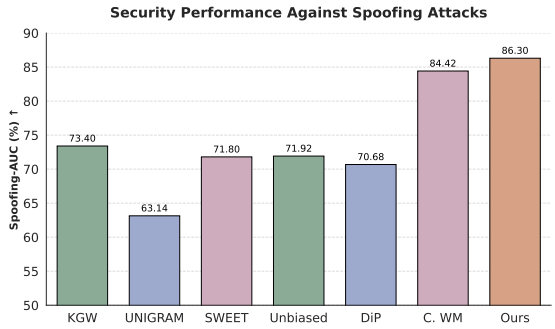

Figure 3: Spoofing-AUC under spoofing attacks.

4.3.3 Security Performance Against Spoofing Attacks

Our security evaluation against spoofing attacks in Figure 3 reveals a significant performance gap between traditional and advanced watermarking schemes. It highlights a critical vulnerability in traditional methods (e.g., KGW), which demonstrate limited robustness by achieving Spoofing-AUC scores only in the 63-74% range. This indicates their poor ability to detect malicious semantic corruption. In stark contrast, our full DFS system maintains a state-of-the-art security posture. Crucially, it not only preserves factual integrity but also enhances security, achieving the highest Spoofing-AUC of 86.30%. This represents a notable improvement over the robust Contrastive WM baseline (84.42%). We attribute this security gain to a direct and beneficial consequence of our fact-protection mechanism. By strengthening the watermark signal around core factual entities, DFS makes the text's semantic fingerprint more robust and resistant to tampering. This result demonstrates a powerful synergy: our approach does not trade security for trustworthiness, but rather leverages fact-preservation to actively bolster security against sophisticated spoofing attacks.

4.3.4 Qualitative Illustration of Spoofing Defense.

Table 2 illustrates our method's defense against complex spoofing attacks that corrupt both facts and sentiment. The baseline system is shown to be vulnerable to an attack that injects negative sentiment (e.g., using "doomed") while simultaneously altering key facts (the year changed to "2016", location to "Indiana"). DFS resists this multi-pronged attack while preserving the original neutral tone and correct data. This ensures that any maliciously altered text, like the one in the example, would fail verification, demonstrating our system's robustness against sophisticated content manipulation.

Method	KCR	Faith.	ROUGE-1	ROUGE-L
Backbone	14.67	74.34	39.91	27.41
+ Stage I	11.90	78.49	38.40	26.28
+ Stage II	8.02	82.01	44.34	31.52
DFS (Stage I + II)	**5.56**	**86.99**	**46.25**	**33.11**

Table 3: Ablation study of the DFS stages with Llama-3.1-8B, demonstrating their synergistic effect.

4.4 Ablation Study

Our ablation study (Table 3) validates the synergistic design of DFS. Stage I provides soft semantic guidance that moderately improves fidelity; Stage II offers stronger, token-level protective bias via the fact-guarded greenlist. Combining both yields the highest Faithfulness (86.99%) and lowest KCR (5.56%), confirming that the integrated approach is essential for achieving substantial improvements in factual integrity.

4.4.1 Impact of Generator Capability

Our results in Figure 4 show that DFS functions as a backbone-agnostic framework, consistently enhancing factual integrity across both Llama3.1-8B

Method Configuration	KCR	Faithfulness	Extracted Entities	ROUGE-1	ROUGE-L
Backbone	14.67	74.34	–	39.91	27.41
Ours (with spaCy)	**5.56**	**86.99**	**1851**	**46.25**	33.11
Ablations on NER Model (within DFS)					
Ours (with bert-ner)	9.24	78.55	1553	44.19	31.10
Ours (with DOUBAO 1.5)	7.13	82.03	1780	45.95	**33.16**

Table 4: Ablation study on NER model selection for Stage II. The choice of model significantly impacts factual preservation and entity coverage, with spaCy providing the most favorable trade-off.

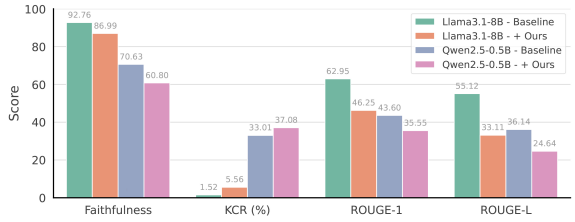

Figure 4: Generalizability of the DFS Algorithm Across Backbone LLMs. DFS consistently improves factual fidelity and text quality when applied to both Llama3.1-8B and Qwen2.5-0.5B models.

and Qwen2.5-0.5B. However, the higher absolute scores on Llama-3.1-8B highlight a critical insight: as a post-hoc framework, DFS relies on the initial generation quality. While DFS effectively shields existing facts, it cannot correct fundamental reasoning errors or hallucinations inherent to weaker backbones (e.g., Qwen2.5-0.5B). Thus, the backbone LLM's capability determines the performance ceiling, confirming that DFS is most effective when paired with a high-quality generator.

4.4.2 Impact of NER Model Choice in Stage II

Table 4 indicates that the Stage II NER model directly affects factual integrity, since it determines which spans are treated as "facts" and thus receive protection. A common failure mode is *missed or fragmented entities*: when key fact-bearing spans (e.g., multi-token entities and number/date expressions) are not extracted as coherent units, they remain unprotected (or only partially protected), which makes them more susceptible to corruption and consequently increases KCR while reducing faithfulness. Compared with BERT-NER and DOUBAO 1.5, whose extractions more often suffer from span incompleteness and less relevant/noisy entities, spaCy produces spans that are both more complete and more aligned with salient facts, and therefore achieves the best overall performance (KCR: 5.56%, Faithfulness: 86.99%).

Model	Method	KCR	Faithfulness
Llama 3.1	SWEET	22.80	68.67
	+ Stage I	20.40	71.05
Qwen 2.5	SWEET	29.54	62.19
	+ Stage I	29.29	63.06

Table 5: Performance of the DFS when combined with an in-processing watermarking method.

4.4.3 Mechanistic Incompatibility with In-Processing Watermarks

To validate the necessity of our co-designed architecture, we integrate DFS with SWEET, a representative in-processing watermark method. Unlike its effectiveness when paired with a post-hoc semantic watermarking backbone, this combination yielded only marginal KCR improvement and slight Faithfulness gains (Table 5), indicating a fundamental mechanistic conflict: post-hoc watermarking provides a stable, complete text for DFS to secure, whereas in-processing methods like SWEET intervene during dynamic generation, where DFS's soft, span-conditioned guidance lacks explicit targets during decoding, since key entities/numbers are not yet formed. Therefore, fact protection and watermarking cannot be simply stacked and must be co-designed, and our results support DFS's synergy with post-hoc semantic rewriting for trustworthy RAG systems.

5 Conclusion

We address faithfulness hallucinations, which are factual corruptions introduced by fact-agnostic watermarking in RAG. We propose DFS, a dual-shield framework that combines evidence-aware semantic guidance with a hard safeguard to enforce knowledge loyalty. Across datasets and backbone LLMs, DFS substantially improves faithfulness and reduces KCR while maintaining strong detectability and robustness, enabling watermarking without sacrificing truth. Future work will extend DFS beyond entities and numbers to relation-bearing spans.

Limitations

While DFS improves factual fidelity for post-hoc watermarking in RAG, it has several limitations. First, DFS may be less reliable on very short texts, where limited token budget makes decoding and payload recovery unstable. Second, under aggressive paraphrasing or high-distortion rewriting, the green-list hit rate can degrade toward a near-random baseline, leading to recovery failures. Third, as a post-hoc framework, DFS largely preserves facts that already appear in the initial generation and therefore cannot remedy fundamental reasoning errors or hallucinations produced by weaker backbones. Fourth, DFS relies on accurate fact identification (e.g., entities and numbers): imperfect extraction can cause under-protection or overly noisy constraints, and different extractors may trade off coverage and fidelity differently. Finally, DFS may not combine cleanly with in-processing watermarking (e.g., logit-biasing methods), since global factual constraints can mechanistically interfere with generation-time interventions, resulting in limited marginal gains.

Acknowledgments

We would like to thank the reviewers in advance for their valuable time and insightful feedback. This work is supported by the Beijing High Innovation Plan (No. 202504841069). We used Gemini for polishing the text.

References

S. Aaronson and H. Kirchner. 2022. Watermarking gpt outputs. https://www.scottaaronson.com/talks/watermark.ppt.

Li An, Yujian Liu, Yepeng Liu, Yang Zhang, Yuheng Bu, and Shiyu Chang. 2025. Defending llm watermarking against spoofing attacks with contrastive representation learning. *Preprint*, arXiv:2504.06575.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable watermarks for language models. *Preprint*, arXiv:2306.09194.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. 2024. On the learnability of watermarks for language models. In *The Twelfth International Conference on Learning Representations*.

Matthew Honnibal and Ines Montani. 2017. spacy: Industrial-strength natural language processing in python. https://spacy.io/.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2024. SemStamp: A semantic watermark with paraphrastic robustness for text generation. 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 4067–4082, Mexico City, Mexico. Association for Computational Linguistics.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2024. Unbiased watermark for large language models. In *The Twelfth International Conference on Learning Representations*.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2024. Who wrote this code? watermarking for code generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4890–4911, Bangkok, Thailand. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024. A semantic invariant robust watermark for large language models. *Preprint*, arXiv:2310.06356.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2024. Dipmark: A stealthy, efficient and resilient watermark for large language models.

Hengyuan Xu, Liyao Xiang, Xingjun Ma, Borui Yang, and Baochun Li. 2024a. Hufu: A modality-agnositc watermarking system for pre-trained transformers via permutation equivariance. *CoRR*, abs/2403.05842.

Xiaojun Xu, Yuanshun Yao, and Yang Liu. 2024b. Learning to watermark llm-generated text via reinforcement learning. *Preprint*, arXiv:2403.10553.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. 2023. Robust multi-bit natural language watermarking through invariant features. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2092–2115, Toronto, Canada. Association for Computational Linguistics.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. 2024. {REMARK-LLM}: A robust and efficient watermarking framework for generative large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1813–1830.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. 2024. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*.

A Dataset Details

A.1 WikiEval

We utilize the WikiEval dataset (Es et al., 2024), a standard benchmark specifically designed for evaluating RAG systems. It consists of question-context-answer triples derived from English Wikipedia. This dataset serves as our primary testbed for evaluating factual consistency and watermark detectability in a controlled environment.

A.2 Wiki-Big

To evaluate our framework's performance on a larger scale and ensure its robustness across diverse topics, we constructed the **Wiki-Big** dataset.

Construction Pipeline. We sampled high-quality articles from the English Wikipedia dump. For each article, we employed an LLM (GPT-4) to generate challenging questions that require reasoning over specific facts within the text. The corresponding paragraphs serve as the ground-truth context. This process ensures that the dataset mimics real-world RAG scenarios where retrieval precision and generation faithfulness are critical.

Statistics. Unlike WikiEval, which is relatively small, Wiki-Big comprises 300 samples, covering a wide range of domains including history, science, pop culture, finance and healthcare. This diversity ensures that our KCR metric is tested against various entity types and factual densities, providing a more rigorous assessment of the watermarking framework's impact.

B Baselines

- **KGW** (Kirchenbauer et al., 2023): A widely-used watermarking method based on a hashed green list.
- **Unigram** (Zhao et al., 2024): A provably robust watermark with a simplified fixed green-red grouping strategy.
- **SWEET** (Lee et al., 2024): A selective watermark that enhances detection and preserves quality by applying watermarks only to high-entropy tokens during generation.

- **Unbiased** (Hu et al., 2024): An unbiased watermark framework eliminates output quality degradation in LLMs by ensuring watermarked text distributions match the original model's output while enabling statistically verifiable detection.

- **DiP** (Wu et al., 2024): A distribution-preserving watermarking framework that maintains text quality through reweighted sampling while enabling API-free detection and provable robustness against token modifications.

- **Contrastive WM** (An et al., 2025): A SOTA secure watermarking algorithm that uses contrastive learning to defend against spoofing attacks. As a post-hoc method, its semantic encoder serves as the backbone adopted by our DFS framework in Stage II. Despite its security prowess, it remains vulnerable to faithfulness hallucinations when used alone, making it an ideal and challenging baseline to demonstrate the effectiveness of our co-designed approach.

C Runtime Analysis

Table 6 reports the average wall-clock time per query on the WikiEval dataset (50 queries).

Method	**Avg. Time (s/query)**
SWEET	7.37
Contrastive WM	10.24
Ours (DFS)	11.77

Table 6: Average runtime per query on WikiEval.

DFS introduces a modest overhead of approximately 15% compared to the Contrastive WM backbone. This additional cost arises from Stage I (prompt augmentation via initial generation and entity extraction) and Stage II (NER-based greenlist expansion). Given the substantial gains in factual fidelity demonstrated in our main experiments, we consider this overhead an acceptable trade-off for trustworthy RAG watermarking.

D Hyperparameter Analysis

We varied the logit-bias strength $\delta \in \{0.3, 0.5, 0.7\}$ on WikiEval with Llama-3.1-8B to examine the trade-off between watermark detectability and factual fidelity. Based on this comparison, we use $\delta = 0.5$ as the default setting in the main experiments.

δ	AUC (↑)	KCR (↓)	Faithfulness (↑)
0.3	84.20	3.81	91.29
0.5 (default)	90.68	5.56	86.99
0.7	97.08	8.04	85.71

Table 7: Effect of the logit-bias strength δ on the trade-off between watermark detectability and factual fidelity on WikiEval with Llama-3.1-8B.