Knowledge Graph-Guided Multi-Hop Probing for Hallucination Detection in Large Language Models

Ushtar Ali[1,2,*,†], Steven Lynden[2], Akiyoshi Matono[2] and Toshiyuki Amagasa[1]

[1]University of Tsukuba, Tsukuba, Japan
[2]National Institute of Advanced Industrial Science and Technology (AIST), Japan

Abstract

Large language models (LLMs) often hallucinate factual information, yet existing benchmarks for evaluating hallucinations are typically static or uniformly sampled, providing limited insight into where such failures are most likely to occur. We address this limitation through a knowledge graph-guided probing framework that performs multi-hop traversal over Wikidata from diverse seed entities and scores extracted triples using four graph-derived signals: entity entropy, property-pair pointwise mutual information (PMI), temporal freshness, and a prominence-recency interaction signal. These signals are aggregated into a unified difficulty score φ, which guides the automatic generation of probing questions at each hop toward challenging facts. Experiments on 300 questions generated from multiple base entities across four open-source LLMs reveal that φ is significantly negatively correlated with factual accuracy across all models (r up to -0.314, $p<0.001$). A further comparison of the top-100 signal-guided questions against a random baseline shows accuracy gaps of up to 20.4 percentage points ($p<0.05$), confirming that the proposed strategy effectively aligns with LLM blind spots. Temporal freshness emerges as the strongest predictor of hallucination, while the domain-level analysis further shows that LLM factual knowledge is unevenly distributed, with hallucinations occurring more frequently in certain domains.

Keywords

LLM Hallucination Detection, Knowledge Graphs, KG-LLM alignment, Evaluation Framework

1. Introduction

Large Language Models (LLMs) have become a fundamental component of modern intelligent systems, powering applications such as conversational assistants, question answering, search, scientific discovery, and decision support [1, 2]. Despite their impressive language understanding and generation capabilities, ensuring the factual reliability of their outputs remains a significant challenge. In particular, LLMs are prone to hallucinations, i.e., generating fluent and plausible responses that are factually incorrect or unsupported by evidence [3]. Such hallucinated responses can propagate misinformation, reduce user trust, and negatively affect downstream applications, particularly in domains where factual correctness is essential, such as fact verification, scientific research, healthcare, and legal decision making [3, 4].

Existing approaches for evaluating factual knowledge in LLM responses, primarily rely on static benchmarks or probing datasets. Benchmarks such as TriviaQA [5], Natural Questions [6], and TruthfulQA [7] assess factual recall using fixed question sets. Although these benchmarks have driven significant progress, they inevitably become outdated and are increasingly susceptible to benchmark contamination as LLMs are trained on continuously expanding web-scale corpora [8]. To address this limitation, recent studies have increasingly incorporated structured knowledge from knowledge graphs (KGs) for fact verification, question answering and fine-tuning tasks.

Knowledge graphs (KGs), such as Wikidata [9], provide explicit entity–relation structures and continuously evolving factual knowledge, making them well suited for factual grounding and hallucination analysis [10, 11]. KG-enhanced retrieval-augmented generation (RAG) approaches, such as GraphRAG [12] and TrumorGPT [13], leverage graph-structured representations to retrieve relevant,

Graph-enhanced LLMs for trustwOrthy Web data management (GLOW), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy
*Corresponding author.
ushtar.ali@kde.cs.tsukuba.ac.jp (U. Ali); steven.lynden@aist.go.jp (S. Lynden); a.matono@aist.go.jp (A. Matono); amagasa@cs.tsukuba.ac.jp (T. Amagasa)
© 2026 Copyright © 2026 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-05

CEUR-WS.org/Vol-4313/glow-paper4.pdf

up-to-date evidence for complex reasoning and factual verification, thereby improving the factual consistency of LLM-generated responses. However, these approaches remain largely reactive, as they verify or ground model outputs only after a query has been issued. Their effectiveness heavily depends on the quality, coverage, and freshness of the retrieved knowledge, and they are susceptible to fact conflicts arising from inconsistent evidence across multiple sources. Moreover, they provide little insight into which facts are inherently more likely to induce hallucinations before generation.

To better understand the factual knowledge encoded within LLMs, recent research has increasingly focused on probing methodologies. Rather than using external knowledge to support answer generation, probing leverages external knowledge sources to automatically construct factual evaluation queries, allowing LLMs to be assessed solely based on their parametric knowledge. This enables scalable evaluation beyond static benchmark datasets while providing insights into the factual capabilities and limitations encoded within the model itself. Among these efforts, KGLens [14] leverages knowledge graphs to automatically generate probing questions and efficiently evaluate the alignment between LLMs and structured factual knowledge, revealing LLM blind spots. It was further supported by Entropy-Guided Probing [15] revealing that graph-derived features, can effectively identify regions of a knowledge graph where LLMs are more likely to hallucinate, suggesting that structured knowledge contains informative signals about model blind spots. However, this work considers only a limited set of structural features and focuses on single-hop probing, leaving unanswered whether richer structural and temporal signals can systematically guide multi-hop probing of hallucination-prone knowledge.

Motivated by these observations, we propose a knowledge graph-guided framework for autonomous multi-hop probing of factual hallucinations in LLMs. Starting from a seed entity, the framework performs a breadth-first traversal over Wikidata forming a subgraph to extract factual triples. Each triple is evaluated using four complementary graph-derived signals–entity entropy, property-pair pointwise mutual information (PMI), temporal freshness, and interaction signal, which collectively estimate factual difficulty. These signals guide the probing process by prioritizing triples that are more likely to expose hallucination-prone regions in the KG, while their connected entities are subsequently used to construct multi-hop reasoning paths. In addition, each triple is assigned a semantic property-domain label from a predefined taxonomy comprising Biographical, Geographic, Political, Academic, Organizational, Cultural, and Recognition categories. These labels enable a domain-level analysis of hallucination behavior across different categories of factual knowledge.

We evaluate our approach using four open-source models on automatically generated single-hop and multi-hop factual questions. Experimental results demonstrate that the proposed knowledge graph signals are significantly associated with hallucination difficulty.Furthermore, multi-hop probing demonstrates the importance of compositional reasoning when assessing factual reliability.

The main contributions of this paper are summarized as follows:

- We propose an autonomous knowledge graph signal-guided framework for systematically probing factual hallucinations in large language models.
- We introduce an annotation-free methodology that automatically generates both single-hop and multi-hop factual questions through knowledge graph traversal.
- We provide a domain-level analysis of factual hallucinations, identifying knowledge domains that consistently expose the blind spots of large language models.
- We design graph-derived structural and temporal signals that estimate factual difficulty prior to evaluation, enabling targeted probing of hallucination-prone knowledge.
- We compare our approach against a random baseline using four open-source LLMs, demonstrating that our method identifies hallucination-prone facts more effectively than random probing.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed framework. Section 4 describes the experimental setup and datasets. Section 5 presents the findings and their analysis, and Section 6 concludes the paper and discusses future research directions.

2. Related Work

2.1. Benchmark-based Factual Evaluation

Factual evaluation aims to assess whether the information generated by an LLM is factually correct. To this end, most existing evaluation approaches rely on manually curated benchmark datasets. TriviaQA [5] and Natural Questions [6] evaluate open-domain question answering using fixed collections of factual queries, while TruthfulQA [7] specifically measures a model's tendency to generate plausible yet factually incorrect responses. Collectively, these benchmarks have become standard resources for evaluating the factual knowledge, recall, and reasoning capabilities of LLMs.

Despite their success, benchmark-based evaluation suffers from several inherent limitations. Since benchmark questions remain fixed after construction, they gradually become outdated as world knowledge evolves. Furthermore, benchmark contamination has become increasingly problematic for modern LLMs trained on massive web-scale corpora. Magar and Schwartz [8] demonstrate that overlap between benchmark data and pre-training corpora can compromise the reliability of benchmark evaluation by encouraging memorization and data exploitation, making it difficult to determine whether high performance reflects genuine factual reasoning or previously memorized knowledge. In addition, these static benchmarks treat all evaluation instances equally and provide little understanding of why certain facts are consistently more challenging than others. These limitations motivate the development of scalable evaluation methodologies capable of adapting to continuously evolving knowledge.

2.2. Knowledge Graphs for Factual Grounding and Verification

Knowledge graphs have emerged as an effective source of structured knowledge for improving the factual reliability of LLMs. Large collaborative knowledge graphs such as Wikidata [9] continuously maintain structured entity–relation information, providing an attractive foundation for factual grounding and verification. Recent studies have incorporated knowledge graphs into retrieval-augmented generation (RAG) systems to provide external evidence during inference. GraphRAG [12] organizes retrieved information into graph-based representations to support structured retrieval and reasoning across large document collections, while TrumorGPT [13] combines graph-enhanced retrieval with domain-specific knowledge graphs for trustworthy fact verification. Beyond retrieval, several approaches verify factual consistency after generation. SelfCheckGPT [16] identifies hallucinations by measuring the consistency of multiple sampled responses, whereas FActScore [17] and FacTool [18] evaluate generated claims using external evidence and specialized verification tools.

Although these approaches have substantially improved factual grounding and post-hoc verification, they remain dependent on the quality and consistency of external knowledge sources. Variations in entity and relation representations, incomplete or outdated knowledge, and conflicting facts across heterogeneous sources can all affect retrieval and verification quality. Moreover, since these methods focus on validating generated responses, they provide limited insight into which facts are intrinsically more likely to induce hallucinations before model inference.

2.3. Knowledge Graph-guided Probing of LLMs

An alternative research direction investigates the factual knowledge encoded within LLM parameters through probing. Unlike retrieval-based approaches, probing uses external knowledge sources to construct evaluation questions while assessing the model solely based on its internal parametric knowledge. This enables systematic analysis of what knowledge an LLM effectively covers without relying on external evidence during answer generation.

LAMA [19] introduced one of the earliest probing bechmark by evaluating encoded factual knowledge in LLMs using manually designed cloze templates. Jiang et al.[20] later showed that the effectiveness of knowledge probing depends heavily on prompt design and introduced mining-based and paraphrasing-based methods to automatically generate diverse prompts, enabling more accurate estimation of the factual knowledge encoded within language models. More recently, KGLens[14] leveraged knowledge

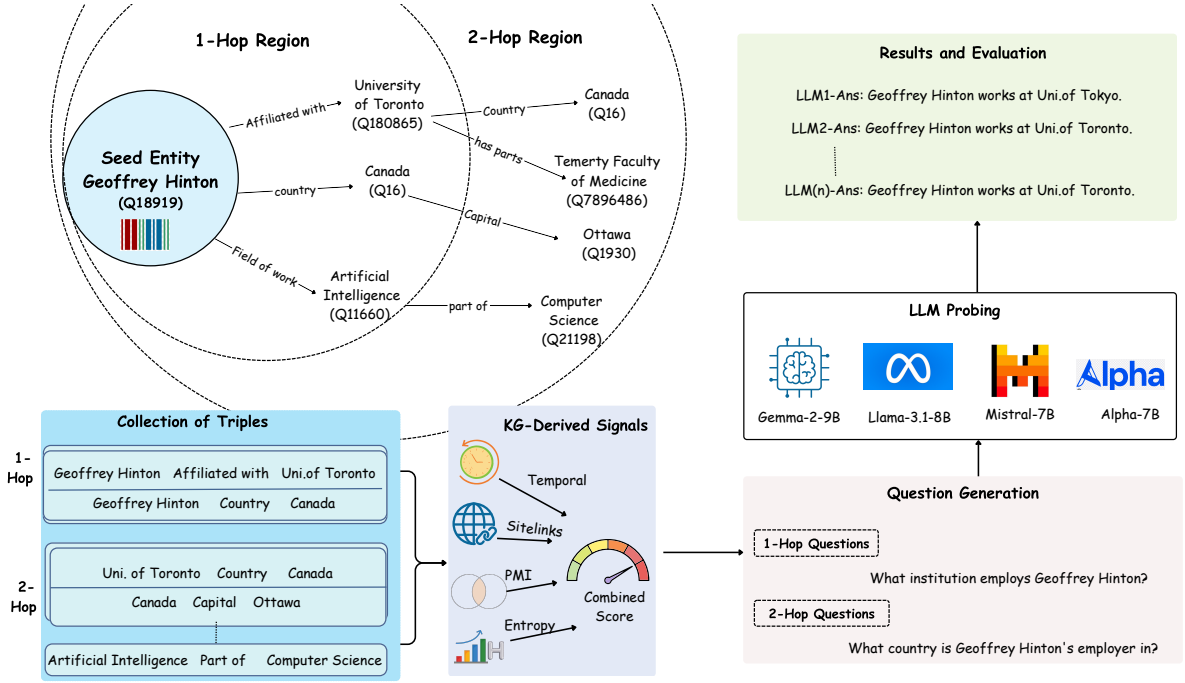


Figure 1: Overview of the proposed knowledge graph-guided multi-hop hallucination probing framework. Starting from a seed entity, the framework incrementally traverses Wikidata, generates KG-guided probing questions, and evaluates LLM factual knowledge against Wikidata ground-truth answers.

graphs to automatically generate probing questions and measure the alignment between LLMs and structured factual knowledge, reducing the dependence on manually constructed templates. Recent work by Ali et al. [15] further demonstrated that structural properties of knowledge graphs, particularly entity entropy, are correlated with LLM factual errors, suggesting that graph-derived signals can reveal regions of knowledge where hallucinations are more likely to occur.

While these methods have substantially advanced factual probing, they largely focus on single-hop factual relations and generally assume that all generated facts are equally informative for evaluation. Consequently, they provide limited guidance on which facts should be prioritized for probing and do not investigate how reasoning over multi-hop knowledge affects hallucination behavior.

3. Methodology

Figure 1 provides an overview of the proposed method which consists of three components: knowledge graph traversal, graph-derived difficulty estimation, question generation and LLM probing. Starting from a seed entity, the framework incrementally traverses Wikidata to construct a local multi-hop knowledge graph. At each hop, graph-derived signals are computed to estimate the factual difficulty of the extracted triples, and the highest-scoring triples are used to guide further graph exploration. This signal-guided traversal focuses the search on regions of the KG that are more likely to expose LLM hallucinations. The extracted triples are subsequently transformed into single-hop and multi-hop probing questions, which are posed to LLMs and evaluated against the corresponding Wikidata ground-truth answers. The details are described below.

3.1. Knowledge Graph Traversal

Given a seed entity e_0, the framework first extracts the one-hop factual statements and constructs a local Wikidata subgraph by performing a breadth-first search (BFS). These factual statements are represented as subject–predicate–object (s, p, o) triples, where s denotes the subject entity, o the corresponding object, and p the property connecting s and o. The resulting set of extracted triples is denoted by $\mathcal{T}$.

Wikidata stores each fact as a statement that may include qualifiers, references, and a validity rank. The framework retains only statements whose object is another Wikidata entity (Q-identifier), ensuring that every extracted fact represents an (s, p, o) relationship. Statements marked as *deprecated* are discarded because they correspond to obsolete or invalid facts, while ontological and maintenance properties (e.g., `P31`, `P279`) are excluded since they describe the schema of the knowledge graph rather than factual knowledge expected to be recalled by an LLM. To control the size of the extracted subgraph, the traversal is governed by three configurable parameters: the maximum number of triples retained per entity (`max_triples_per_entity`), the maximum number of neighboring entities expanded at each hop (`max_entities_per_hop`), and the maximum traversal depth (`max_hops`). In our experiments, these parameters are set to 25, 10, and 2, respectively. Together, they bound the search space while allowing the framework to scale to larger subgraphs, deeper traversals, or multi-seed exploration without modifying the underlying traversal algorithm.

In addition to the extracted triples, the framework collects auxiliary data required for subsequent signal computation. For each entity, the total number of distinct factual property types is retrieved to capture its overall property diversity, independent of the sampled subgraph. Temporal information is obtained exclusively from statement-level qualifiers, namely `P580` (start time) and `P585` (point in time), as these indicate when a specific fact became valid. Entity-level modification timestamps are intentionally excluded because they reflect arbitrary edits to an entity page, such as typo corrections or metadata updates, rather than changes to the underlying fact. Entity labels are resolved through batched API requests and cached locally to eliminate redundant lookups. Finally, each triple is assigned a semantic property-domain label using a predefined taxonomy comprising the Biographical, Geographic, Organizational, Political, Academic, Cultural, and Recognition categories, while all remaining properties are assigned to an *Other* category. This auxiliary data is subsequently used for graph signal computation and domain-level analysis.

3.2. Graph-derived Difficulty Estimation

Difficulty signals are computed iteratively as the knowledge graph expands. At each hop, the extracted triples are scored using the four graph-derived signals described below, and the highest-scoring triples guide the next-hop expansion through their object entities. The resulting scores are used for triple selection and question generation. The individual signals are defined below.

Entropy Score. The entropy score measures the factual diversity associated with a triple by considering the diversity of both its subject and object entities. For each entity e, let $K(e)$ denote the total number of distinct factual property types associated with e in Wikidata. Assuming a uniform distribution over these property types, the entity entropy is computed as

$$H(e) = \log_2 K(e). \tag{1}$$

The entropy score of a triple is then computed by combining the normalized entropy of its subject and object entities:

$$\mathrm{ent}(s, o) = \begin{cases} \frac{1}{2}\left(\hat{H}(s) + \hat{H}(o)\right), & \text{if } \hat{H}(s), \hat{H}(o) > 0, \\ \max\left(\hat{H}(s), \hat{H}(o)\right), & \text{otherwise,} \end{cases} \tag{2}$$

where $\hat{H}(\cdot)$ denotes min–max normalization over all entities in the extracted subgraph. Using the complete set of Wikidata properties rather than only the sampled triples ensures that the entropy score reflects the true semantic diversity of each entity. The normalized entropy values are recomputed after graph expansion so that the final scores capture the complete relational context of the multi-hop subgraph.

PMI Score. The Pointwise Mutual Information (PMI) score measures the strength of association between pairs of properties that co-occur for the same entity [21]. Formally, the PMI between two properties p_i and p_j is defined as

$$\mathrm{pmi}(p_i, p_j) = \log_2 \frac{P(p_i, p_j)}{P(p_i)P(p_j)}. \tag{3}$$

The probabilities are estimated from property co-occurrence statistics computed over the extracted subgraph. For each entity, the maximum PMI value among all property pairs is retained and subsequently normalized. Higher PMI scores correspond to uncommon property combinations, indicating facts that are more specific and therefore less likely to be reinforced by the statistical patterns observed during LLM pre-training.

Temporal Score. The temporal score captures the recency of a fact based on statement-level qualifier timestamps. Specifically, qualifiers `P580` (start time) and `P585` (point in time) are used because they indicate when a fact became valid. Entity-level modification timestamps are deliberately excluded, as they may reflect unrelated edits that do not alter the factual content. Let T denote the year in which evaluation is conducted, and t be the year extracted from statement qualifier. The temporal score is defined as:

$$\mathrm{temp}(t) = \begin{cases} 1.0, & t \geq T_e, \\ 0.6, & t = T_e - 1, \\ 0.3, & t = T_e - 2, \\ 0.0, & \text{otherwise or absent.} \end{cases} \tag{4}$$

This scoring emphasizes facts introduced close to the time of evaluation, as they are less likely to have been incorporated into the training data of contemporary LLMs and are therefore more susceptible to hallucination. The score differentiates only the most recent years because this period is most relevant to the knowledge cut-off of current LLMs, while older facts are generally expected to be represented regardless of their exact publication year.

Sitelinks Score. The sitelinks score, denoted by $\mathrm{sl}(e)$, is computed as the min–max normalized number of Wikipedia language editions containing an article for each unique entity e in the extracted subgraph. It serves as a measure of an entity's popularity, as entities covered by more Wikipedia language editions are generally more well-known and are therefore more likely to be represented in LLM training data.

Interaction Score. The interaction score captures the combined effect of entity's popularity and temporal recency. It is defined as

$$\mathrm{inter}(s, o, t) = \overline{\mathrm{sl}}(s, o) \times \mathrm{temp}(t), \qquad \overline{\mathrm{sl}}(s, o) = \frac{\mathrm{sl}(s) + \mathrm{sl}(o)}{2} \tag{5}$$

where $\overline{\mathrm{sl}}(s, o)$ denotes the average normalized sitelinks score of the subject entity s and object entity o, providing a triple-level measure of entity prominence analogous to the entropy score, and $\mathrm{temp}(t)$ is the temporal score of the corresponding fact. Higher interaction scores correspond to triples involving prominent entities and recently established facts, making them more challenging for LLMs because knowledge acquired during pre-training may no longer reflect the most up-to-date information.

Combined Difficulty Score. The four graph-derived signals are aggregated into a unified score, φ, which is used to rank triples during graph-guided traversal and question generation:

$$\varphi(s, p, o) = \frac{1}{4}\big(\mathrm{ent}(s, o) + \mathrm{pmi}(s) + \mathrm{temp}(t) + \mathrm{inter}(s, o, t)\big) \tag{6}$$

All signals are normalized to the range $[0, 1]$ before aggregation, and equal weights are assigned since no signal is prioritized. Higher values of φ indicate triples with greater estimated factual difficulty.

3.3. Question Generation and LLM Probing

We transform the selected triples into natural language questions for probing the factual knowledge encoded within LLMs. To minimize the influence of a particular generation strategy, the framework employs two complementary question generation methods: an LLM-based generator and a fine-tuned T5 model [1]. Both methods generate single-hop and multi-hop questions (see Appendix A). Single-hop questions assess the recall of individual facts represented by a single triple, whereas multi-hop questions require reasoning over two connected triples through an intermediate entity. This design enables the evaluation of both factual recall and compositional reasoning while assessing the robustness of the proposed framework across different question generation strategies. The details are as follows:

Single-hop Questions. For each of the top-K hop-1 triples selected by the combined difficulty score ϕ, both question generation methods independently produce a question whose correct answer is the object entity o. The LLM-based method takes the verbalized triple as input and generates a factual question about subject entity s, whereas the T5-based method is conditioned on the answer together with a context sentence derived from the same triple. For example, given the triple (*Geoffrey Hinton*, *employer*, *University of Toronto*), the LLM-based method generates *"What institution employs Geoffrey Hinton?"*, while the T5-based method generates *"What is Geoffrey Hinton's employer?"*

Since Wikidata allows multiple valid values for the same subject–property pair, the gold answer set is defined as the complete set of valid object entities associated with (s, p) in $\mathcal{T}$:

$$A^*(s, p) = \{ o \mid (s, p, o) \in \mathcal{T} \}. \tag{7}$$

Constructing the gold answer set before triple selection ensures that all valid answers are retained during evaluation, regardless of which triples are ultimately selected for probing.

Multi-Hop Questions. To evaluate compositional reasoning, the framework generates multi-hop questions from two-hop paths discovered during the signal-guided graph traversal. Each path has the form

$$e_1 \xrightarrow{p_1} e_2 \xrightarrow{p_2} e_3,$$

where the intermediate entity e_2 connects the two facts. During question generation, this intermediate entity is intentionally omitted, requiring the model to infer it before predicting the final answer e_3. For example, the path (*Geoffrey Hinton* → *employer* → *University of Toronto* → *country* → *Canada*) is transformed into the question *"What country is Geoffrey Hinton's employer in?"*, where *University of Toronto* must be inferred implicitly. The corresponding gold answer set contains all valid terminal entities e_3 associated with the path (e_1, p_1, e_2, p_2) in $\mathcal{T}$.

Each generated question is prompted to four open-source LLMs of varying capability in a zero-shot setting (see Appendix B). Consequently, the responses rely solely on the model's parametric knowledge. To ensure consistency, each question is evaluated three times using low-temperature sampling ($T = 0.1$), so that disagreement across probes reflects genuine uncertainty rather than stochastic variation. The majority response is selected as the final answer.

4. Experiments and Evaluation

Knowledge Graph and Dataset. We construct a 2-hop Wikidata subgraph using KG-guided traversal (as described in Section 3). The traversal is initialized from 10 seed entities selected from diverse

[1] https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap

Table 1
Overall factual accuracy of the evaluated LLMs on the probing set, LLM-based and T5-based columns report accuracy on questions generated by LLaMA and T5 respectively; Combined averages across all 300 questions.

Model	Family	Params.	LLM-based	T5-based	Combined
Gemma-2-9B	Gemma 2	9B	60.7%	55.3%	**58.0%**
Llama-3.1-8B	Llama 3.1	8B	55.3%	49.3%	52.3%
Mistral-7B	Mistral	7B	48.0%	40.7%	44.3%
Alpha-7B	StableLM	7B	6.0%	6.0%	6.0%

domains, including science, politics, business, sports, and philosophy. At hop 1, the top-$K = 100$ triples are selected, while at hop 2, the top-$K = 50$ triples ranked by combined difficulty score φ are selected, resulting in 150 scored triples. These scored triples were used to generate 300 probing questions comprising 100 LLM-generated, 100 T5-generated, 50 LLM-based multi-hop, and 50 T5-based multi-hop questions as described in Section 3.

Models. We employ five models for question generation and probing. All these models are publicly available on Hugging Face [2]. The evaluated models are summarized below.

- **Llama-3.1-8B:** `Llama-3.1-8B-Instruct`: An 8-billion-parameter instruction-tuned model from the Llama, designed for general-purpose language understanding and generation [22].
- **T5:** `mrm8488/t5-base-finetuned-question-generation-ap`: A T5-base model fine-tuned for answer-aware question generation.
- **Gemma-2-9B:** `Gemma-2-9B-IT`: A 9-billion-parameter instruction-tuned model developed by Google, optimized for efficient and high-quality instruction following [23].
- **Mistral-7B:** `Mistral-7B-Instruct-v0.3`: A 7-billion-parameter instruction-tuned model developed by Mistral AI for reasoning and conversational tasks [24].
- **Alpha-7B:** `StableLM-Tuned-Alpha-7B`: A 7-billion-parameter instruction-tuned model developed by Stability AI for general-purpose text generation and conversational tasks.

For question generation, we use Llama-3.1-8B and the fine-tuned T5 model to generate both single-hop and multi-hop probing questions. For LLM probing, we evaluate four instruction-tuned models: Llama-3.1-8B, Gemma-2-9B, Mistral-7B, and Alpha-7B.

Evaluation. Model responses are compared against the gold-answer set using cosine similarity between Sentence-BERT (all-MiniLM-L6-v2 [3]) embeddings. A response is considered correct if its maximum similarity to any gold-answer exceeds a threshold of $\tau = 0.65$. Additionally, to validate the selection of $\tau = 0.65$, we randomly sampled 20 responses across all four models and compared the cosine-similarity labels with independent evaluations from GPT-4o [4]. Both methods agreed on 80% (16/20) of samples, supporting the reliability of the selected threshold for large-scale evaluation.

5. Results

Table 1 reports the overall factual accuracy of the four evaluated models on T5-generated questions, LLM-generated questions, and the combined question set. Since both question generation methods are derived from the same underlying KG triples, they assess the same factual knowledge through different linguistic formulations. We therefore report the combined accuracy throughout the paper. Gemma-2-9B achieves the highest combined accuracy (58.0%), followed by Llama-3.1-8B (52.3%) and Mistral-7B (44.3%) while Alpha-7B performs substantially lower (6.0%). Table 2 presents the correlation between proposed graph-guided signals and factual correctness of the models. The combined difficulty

[2] https://huggingface.co
[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[4] https://openai.com/index/hello-gpt-4o/

Table 2
Pearson correlation between Graph-guided signals and model's factual responses.

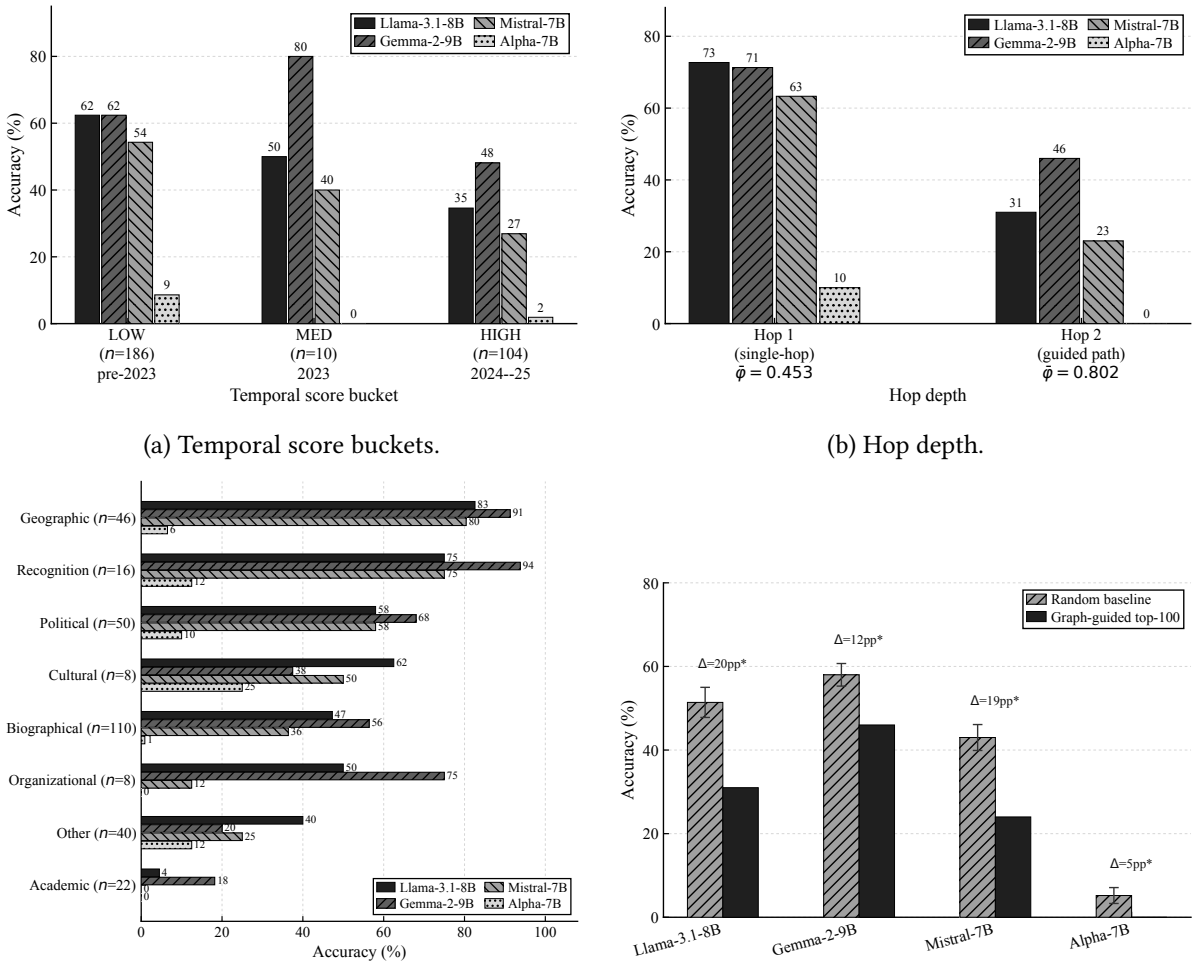
Signal	Llama-3.1-8B	Gemma-2-9B	Mistral-7B	Alpha-7B
Entropy	+0.082 n.s.	+0.010 n.s.	$+0.121^{*}$	−0.019 n.s.
PMI	+0.069 n.s.	−0.001 n.s.	$+0.101^{\dagger}$	−0.004 n.s.
Temporal	-0.252^{***}	-0.119^{*}	-0.247^{***}	-0.124^{*}
Sitelinks	+0.016 n.s.	+0.015 n.s.	+0.087 n.s.	−0.051 n.s.
Interaction	-0.130^{*}	−0.030 n.s.	-0.135^{*}	−0.032 n.s.
Combined	$\mathbf{-0.314}^{***}$	$\mathbf{-0.173}^{**}$	$\mathbf{-0.288}^{***}$	$\mathbf{-0.183}^{**}$

$n = 300$ per model. $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, $^{\dagger}p < 0.1$; n.s. = not significant.

score φ is significantly negatively correlated with correctness for all four models, indicating that facts with higher difficulty scores are more hallucination-prone and more likely to be answered incorrectly. The strongest correlations are observed for Llama-3.1-8B (r=−0.314, p<0.001) and Mistral-7B (r=−0.288, p<0.001), followed by Alpha-7B (r=−0.183, p<0.01) and Gemma-2-9B (r=−0.173, p<0.01). Among the individual signals, the temporal score is the most consistent predictor, showing significant negative correlations across all models. This suggests that recently established facts are a major source of factual errors. The interaction score is also significant for Llama-3.1-8B (r=−0.130, p<0.05) and Mistral-7B (r=−0.135, p<0.05), indicating that prominent entities associated with recent facts are particularly challenging. Figure 2a reports accuracy across temporal score buckets. Accuracy decreases from LOW buckets containing stable pre-2023 facts to HIGH buckets containing recently established facts from 2024–25. The decline is most pronounced for Llama-3.1-8B and Mistral-7B, with drops of 27.8 and 27.4 percentage points, respectively. This trend supports the role of fact recency as a key source of factual difficulty. Figure 2b further evaluates the effect of signal-guided traversal across hop depths. Hop-2 triples have the highest mean difficulty score ($\bar{\varphi}$=0.802) and lower accuracy than hop-1, showing that the guided expansion prioritizes more challenging facts as the graph grows. In contrast, Hop-1 triples are answered more accurately, indicating that directly connected facts around the seed entities are generally easier for the evaluated models. Figure 2c reports accuracy by property domain. Academic properties, such as field of work, doctoral supervisor, and academic degree, are the most challenging across models, with particularly low accuracy for Llama-3.1-8B (4.5%) and Mistral-7B (0.0%). This suggests that fine-grained scholarly facts are weakly represented in the parametric knowledge of the evaluated LLMs. Geographic facts are answered most reliably, likely because they are stable, common, and well represented in training data.

To further validate whether graph-guided selection targets genuinely harder questions, we compare the top-100 questions ranked by φ against a random baseline of 100 question. To reduce sampling variability, the random selection was repeated 20 times, and the average performance was used as the baseline. As shown in Figure 2d, graph-guided selection consistently yields lower accuracy across all models, with gaps of 20.4, 12.0, and 18.9 percentage points for Llama-3.1-8B, Gemma-2-9B, and Mistral-7B respectively, all statistically significant (p<0.05). This confirms that the combined signal score reliably identifies hallucination-prone facts that are less likely to be captured by random sampling.

Overall, the results show that the proposed graph signals effectively align with hallucination-prone regions in the KG, revealing the factual blind spots of LLMs. Recent facts consistently lead to lower accuracy, especially when they involve prominent entities whose parametric representations may be outdated. Similarly, hop-2 questions expose relational weaknesses by requiring models to recover second-order factual links rather than direct seed-entity facts. Domain-level results further show that academic properties remain consistently difficult, suggesting limited coverage of fine-grained scholarly knowledge in pre-training data. The consistent significance of φ across models indicates that the combined score captures these regions of factual difficulty in a model-independent manner.



(a) Temporal score buckets.

(b) Hop depth.

(c) Property domains.

(d) Signal-guided vs. random.

Figure 2: Result analysis of signal-guided probing across temporal score buckets, hop depth, property domains, and random baseline comparison.

6. Conclusion and Future Work

We presented a KG-guided probing framework that uses graph-derived signals for multi-hop question generation aligned with LLM blind spots. The combined difficulty score φ consistently identifies challenging facts, while graph-guided selection produces harder probes than random sampling. The analysis further reveals higher hallucination risk for recent facts and consistent weaknesses in academic-related properties across models.

The temporal score assigns 0.0 to statements without P580/P585 qualifiers, thereby treating missing temporal annotations similarly to older facts. Since qualifier coverage varies across Wikidata properties and domains, this may introduce domain-dependent bias and contribute to observed performance differences. Moreover, the current study is limited to a relatively limited set of entities and a small portion of the graph.

Future work will scale the evaluation across larger entity sets and datasets, explore deeper hops and other knowledge graphs such as DBpedia and Bio2RDF, compare against baselines such as KGlens [14], and examine finer-grained failure categories beyond binary correctness. We will also investigate missingness-aware and domain-sensitive temporal scoring to distinguish unavailable temporal information from genuinely older facts.

7. Acknowledgements

This paper is based on the results obtained from the project, “Research and Development Project of the Enhanced infrastructures for Post-5G Information and Communication Systems” (JPNP20017), commissioned by the New Energy and Industrial Technology Development Organization (NEDO) and JST CREST Grant Number JPMJCR22M2.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-5.4 to assist in experimental code and implementation. Additionally, GPT-5 was used for grammar correction, language editing and improving the readability. All AI-generated content were reviewed and validated by the authors as needed and take(s) full responsibility for the publication’s content.

References

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258 (2021).

[2] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, arXiv preprint arXiv:2303.18223 1 (2023) 1–124.

[3] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, ACM computing surveys 55 (2023) 1–38.

[4] J. Maynez, S. Narayan, B. Bohnet, R. McDonald, On faithfulness and factuality in abstractive summarization, in: Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 1906–1919.

[5] M. Joshi, E. Choi, D. S. Weld, L. Zettlemoyer, Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1601–1611.

[6] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al., Natural questions: a benchmark for question answering research, Transactions of the Association for Computational Linguistics 7 (2019) 453–466.

[7] S. Lin, J. Hilton, O. Evans, Truthfulqa: Measuring how models mimic human falsehoods, in: Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers), 2022, pp. 3214–3252.

[8] I. Magar, R. Schwartz, Data contamination: From memorization to exploitation, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2022, pp. 157–165.

[9] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Communications of the ACM 57 (2014) 78–85.

[10] J. Z. Pan, S. Razniewski, J.-C. Kalo, S. Singhania, J. Chen, S. Dietze, H. Jabeen, J. Omeliyanenko, W. Zhang, M. Lissandrini, et al., Large language models and knowledge graphs: Opportunities and challenges, arXiv preprint arXiv:2308.06374 (2023).

[11] G. Agrawal, T. Kumarage, Z. Alghamdi, H. Liu, Can knowledge graphs reduce hallucinations in llms?: A survey, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 3947–3960.

[12] M. Barry, G. Caillaut, P. Halftermeyer, R. Qader, M. Mouayad, F. Le Deit, D. Cariolaro, J. Gesnouin, Graphrag: leveraging graph-based efficiency to minimize hallucinations in llm-driven rag for finance data, in: Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK), 2025, pp. 54–65.

[13] C. N. Hang, P.-D. Yu, C. W. Tan, Trumorgpt: Graph-based retrieval-augmented large language model for fact-checking, IEEE Transactions on Artificial Intelligence (2025).
[14] S. Zheng, H. Bai, Y. Zhang, Y. Su, X. Niu, N. Jaitly, Kglens: Towards efficient and effective knowledge probing of large language models with knowledge graphs, arXiv preprint arXiv:2312.11539 (2023).
[15] U. Ali, S. Lynden, A. Matono, T. Amagasa, Entropy-guided probing for predicting llm hallucinations with knowledge graph features, in: International Conference on Database and Expert Systems Applications, Springer, 2025, pp. 68–82.
[16] P. Manakul, A. Liusie, M. Gales, Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, in: Proceedings of the 2023 conference on empirical methods in natural language processing, 2023, pp. 9004–9017.
[17] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. Koh, M. Iyyer, L. Zettlemoyer, H. Hajishirzi, Factscore: Fine-grained atomic evaluation of factual precision in long form text generation, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 12076–12100.
[18] I. Chern, S. Chern, S. Chen, W. Yuan, K. Feng, C. Zhou, J. He, G. Neubig, P. Liu, et al., Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios, arXiv preprint arXiv:2307.13528 (2023).
[19] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), 2019, pp. 2463–2473.
[20] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How can we know what language models know?, Transactions of the Association for Computational Linguistics 8 (2020) 423–438.
[21] K. Church, P. Hanks, Word association norms, mutual information, and lexicography, Computational linguistics 16 (1990) 22–29.
[22] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).
[23] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, et al., Gemma 2: Improving open language models at a practical size, arXiv preprint arXiv:2408.00118 (2024).
[24] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

A. Question Generation Prompts

```
Single-hop prompt: Fact: {subject} — {property} — {value}
Write a single factual question about "{subject}" whose correct
answer is exactly "{value}". Return only the question, no explanation.
Multi-hop Prompt: Fact 1: {e1} — {p1} — {e2} Fact 2: {e2} — {p2} — {e3}
Write a single factual question about "{e1}" whose correct answer
is exactly "{e3}". The question must require knowing both facts.
Return only the question, no explanation.
```

B. Probing Prompt

```
Answer the following factual question with a short, precise answer.
Do not add explanations or extra text.
  Question: {question} Answer:
```