KGHaluBench: A Knowledge Graph-Based Hallucination Benchmark for Evaluating the Breadth and Depth of LLM Knowledge

Alex Robertson[1] **Huizhi Liang**[1] **Mahbub Gani**[2]
Rohit Kumar[2] **Srijith Rajamohan**[2*]
[1]School of Computing, Newcastle University [2]Sage Ai, Sage Group PLC
{a.robertson4, huizhi.liang}@ncl.ac.uk
{mahbub.gani, rohit.kumar2, srijith.rajamohan}@sage.com

Abstract

Large Language Models (LLMs) possess a remarkable capacity to generate persuasive and intelligible language. However, coherence does not equate to truthfulness, as the responses often contain subtle hallucinations. Existing benchmarks are constrained by static, narrow questions, resulting in limited coverage and misleading evaluations. We present **KGHaluBench**, a **K**nowledge **G**raph-based **hallu**cination **bench**mark that assesses LLMs across the breadth and depth of their knowledge, providing a fairer and more comprehensive insight into LLM truthfulness. Our framework utilises the KG to dynamically construct challenging, multifaceted questions, whose difficulty is then statistically estimated to address popularity bias. Our automated verification pipeline detects abstentions and verifies the LLM's response at both conceptual and correctness levels to identify different types of hallucinations. We evaluate 25 frontier models, using novel accuracy and hallucination metrics. The results provide a more interpretable insight into the knowledge factors that cause hallucinations across different model sizes. KGHaluBench is publicly available[1] to support future developments in hallucination mitigation.

1 Introduction

Recent advancements in language generation and critical reasoning of Large Language Models (LLMs) have driven their widespread adoption (Minaee et al., 2025), with their remarkable capabilities revolutionising domains such as science, healthcare, and finance (Kaddour et al., 2023; Busch et al., 2025; Li et al., 2024). However, a detrimental yet inevitable limitation with LLMs lies in their tendency to generate inaccurate or misleading content, often referred to as 'hallucinations' (Xu et al., 2024). These may diverge from the intent of the user's input, contradict previously established outputs, and/or conflict with verifiable factual knowledge (Zhang et al., 2023), undermining the LLM's trustworthiness, interpretability, and factual accuracy (Narayanan Venkit et al., 2024). As more proficient LLMs generate increasingly convincing outputs, users could perceive these responses as the truth, amplifying the potential consequences of hallucinations. This emphasises the need for robust benchmarks to quantify hallucinations and assess the effectiveness of mitigation strategies.

The breadth of knowledge assessed in current benchmarks is often constrained by their static nature. This flaw often limits their ability to reflect realistic accuracy and hallucination rates. Some pre-constructed QA benchmarks capture the knowledge before and surrounding the time of their release (Berant et al., 2013; Bordes et al., 2015; Kwiatkowski et al., 2019), but are unable to evaluate a model's performance on newly emerging facts and information. Incorporating a knowledge graph (KG) can alleviate these limitations; however, several KG-based benchmarks revert to static datasets for dynamic generation, thereby limiting the range of assessment and failing to reflect the diversity of knowledge users' queries. (Zhu et al., 2024; Sun et al., 2024; Dammu et al., 2025).

The depth of knowledge is rarely probed beyond surface-level details due to the question style used in the benchmarks. Often, closed or multiple-choice questions are favoured over open-ended questions (Hendrycks et al., 2021; Rahman et al., 2024), as accurately evaluating long-form responses is a current research challenge. However, this question style is knowledge-restrictive, limiting the LLM to choose from a specific set of answers. Instead, some benchmarks use simple questions (Bordes et al., 2015; Joshi et al., 2017), which are typically short, open-ended queries with a single, verifiable answer. These questions require

[*]Current affiliation: Redis
[1]Code available at https://github.com/c0037654Newcastle/KGHaluBench

Findings of the Association for Computational Linguistics: EACL 2026, pages 3975–3989
March 24-29, 2026

the LLM to draw on its internalised representations, but are unable to capture multiple elements of deeper knowledge within a single question.

To address the limitations above, we present KGHaluBench, a **KG**-synergised **hallu**cination **bench**mark. We leverage the comprehensive information within a KG to assess both factual accuracy and hallucination rate across an LLM's knowledge. Our benchmark includes a dynamic question-generation module that draws on the KG to retrieve a random selection of entities spanning diverse topics. With an entity as the focal point, we generate an open-ended compound question that targets three aspects of information, both activating and assessing the depth of an LLM's knowledge. Our response verification framework assesses the factuality of long-form text by identifying hallucinations in the LLM's output. The framework includes an abstention filter to detect expressions of uncertainty, an initial entity-level filter to identify semantic misalignment with the entity, and a final fact-level check to verify correctness against grounded facts. We introduce novel metrics to evaluate our extensive experimental results, Weighted accuracy, W_{a}, scales standard accuracy based on the estimated difficulty of the content within the assessment, providing a fairer measure of performance. Breadth of knowledge and depth of knowledge hallucination rates, Halu_{BOK} and Halu_{DOK}, are derived from what stage in the response verification framework the hallucination was detected, offering an insight into which aspects of the LLM's knowledge caused the hallucination. Our key contributions are:

- We propose a question-generation approach that leverages the KG's relational structure to formulate compound questions over dynamically selected entities.
- We construct an automated verification framework with coarse entity-level and fine-grained fact-level filters to assess the factuality of the LLM's output, achieving 79.19% and 87.74% agreement with human judgment, respectively.
- We develop a statistical method to assess the difficulty of an assessment and scale the accuracy accordingly, generating a novel metric, W_{a}.
- We conduct an extensive experiment using 25 open-source and proprietary LLMs, leveraging Halu_{BOK} and Halu_{DOK} to identify factors in LLMs' knowledge that may cause hallucinations.

2 Related Work

QA Benchmarks: The Question Answer (QA) benchmarks typically evaluate LLMs through open-ended generation, with results assessed by metrics or automated judges. SimpleQA (Wei et al., 2024) utilises short, fact-seeking questions with a single, verifiable answer. While HotpotQA (Yang et al., 2018) introduces additional complexity, requiring multi-hop reasoning to arrive at the answer. These static QA benchmarks quickly become outdated, failing to challenge the constantly advancing LLMs. While dynamic QA benchmarks such as FreshQA (Vu et al., 2023) and RealTimeQA (Kasai et al., 2024) exist, they are challenging to maintain due to complicated data management and frequent updates. Knowledge-Graph Question-Answer (KGQA) benchmarks address this by using KGs, such as Wikidata and DBpedia (Vrandečić and Krötzsch, 2014; Auer et al., 2007), to generate questions. While classic KGQA benchmarks such as ComplexWebQuestions (Talmor and Berant, 2018) and FreebaseQA (Jiang et al., 2019) are static, using fixed KG snapshots, GraphEval (Liu et al., 2024) and Dynamic-KGQA (Dammu et al., 2025) dynamically construct test datasets to remain up-to-date. However, random sampling from the KG may introduce an entity-popularity bias, leading to assessments dominated by well-known entities. To mitigate this, some KGQA benchmarks incorporate discrete difficulty levels. KG-FPQ (Zhu et al., 2024) generates confusability levels based on the editing methods, whereas Head-To-Tail (Sun et al., 2024) calibrates difficulty based on the entity popularity. Yet discrete difficulties can create fairness issues when questions of varying difficulty are grouped. Therefore, KGHaluBench statistically estimates the difficulty of each question, aggregates for the assessment, and scales the accuracy accordingly, ensuring reliable evaluation regardless of the assessment's content.

Hallucination Benchmarks: Numerous benchmarks have been developed to assess an LLM's tendency to generate plausible but factually unsupported content. HalluLens (Bang et al., 2025) separates hallucinations from fact by evaluating the LLM's response to extrinsic and intrinsic hallucinations. TruthfulQA (Lin et al., 2022) focuses on response truthfulness using questions replicating false beliefs or misconceptions. HaluEval (Li et al., 2023) uses hallucinated samples to evaluate an LLM's ability to detect hallucinations. How-

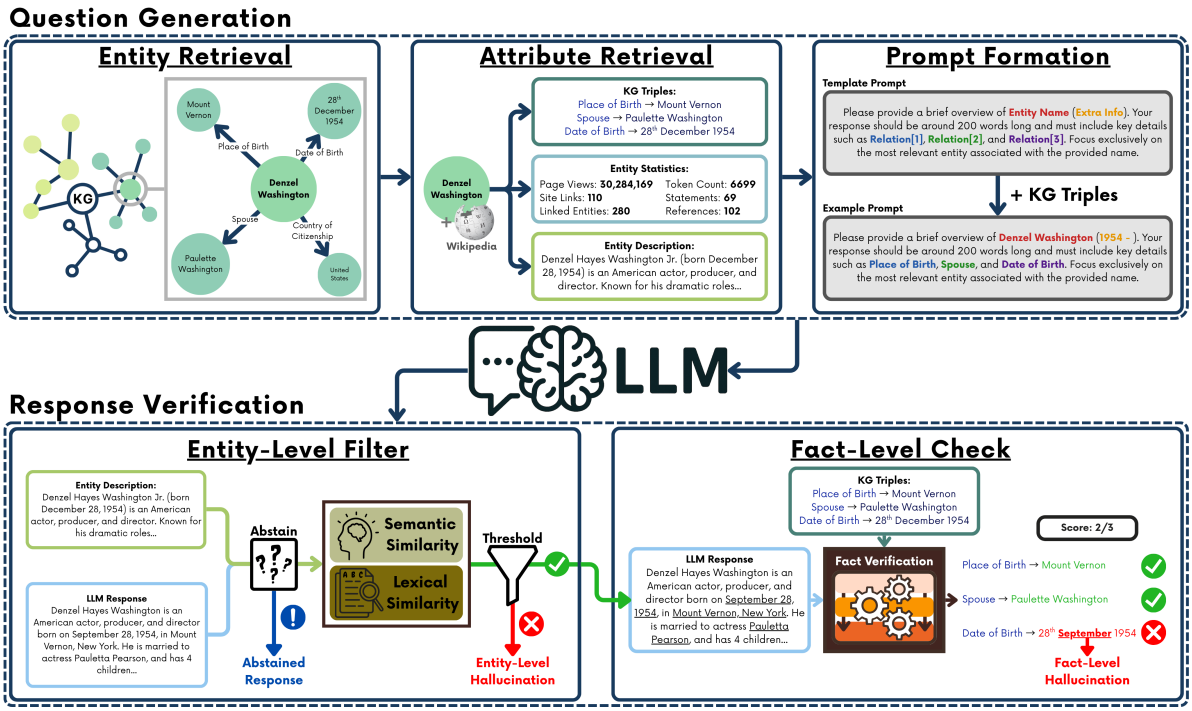

Figure 1: Framework of the KGHaluBench.

ever, many benchmarks rely on one-dimensional metrics, such as Accuracy, Accept/Refusal rates, BLEU, and BERTScore, which limit the interpretability of the results and leave the underlying causes of LLMs' performance unclear. To address this, benchmarks utilise more advanced verification approaches. FEVER (Thorne et al., 2018) uses a Natural Language Inference (NLI) model to evaluate if the response contains, contradicts, or does not mention the evidence. FactCC (Kryscinski et al., 2020) uses a BERT-based model to verify the factual overlap between a response and the evidence. FactScore (Min et al., 2023) decomposes a generation into atomic facts to calculate the amount which are supported by evidence. However, these approaches still collapse results into aggregate metrics that fail to provide clear, actionable insights. KGHaluBench instead decomposes the common hallucination rate into Halu_{BOK} and Halu_{DOK} to determine the knowledge level responsible for the hallucination.

3 The KGHaluBench Benchmark

We propose KGHaluBench, a novel benchmark for evaluating the truthfulness of LLMs when answering challenging questions that span the breadth and depth of their knowledge. As shown in Figure 1, the benchmark consists of two complementary components. The *Question Generation Module* extracts a random entity (*e.g. Denzel Washington*) from the KG to become the centre of the question. The KG's structure, together with external databases, is then leveraged to fetch the entity's *KG Triples, Statistics*, and *Description* needed to construct and validate the multifaceted question. The *Response Verification Module* employs a two-layer framework that first checks the *LLM's response* against the entity's *Wikipedia Description* to ensure non-abstention and confirm a basic understanding of the focal entity (*e.g. Denzel Washington*). Non-hallucinated responses are then verified at the fact level by comparing their claims to the *KG Triples* to ensure correctness.

3.1 Question Generation

The *Question Generation Module* dynamically constructs coherent and challenging questions utilising the KG to alleviate the static, pre-constructed nature of current QA benchmarks.

3.1.1 Entity Retrieval:

For each question, we first define a focal entity. Using a batch call to the KG, we retrieve a random sample of entities, recording each ID and corresponding type. Invalid types are filtered against a predefined list of valid entity types, categorised by their KG frequency as Very Common *(e.g., Human)*, Common *(e.g., Business)*, or Uncommon *(e.g., Painting)*. We order the sampling prioritising

less common entities to maintain a balanced distribution of entity types, ensuring the benchmark remains topically broad yet structured. Table 2 illustrates the resulting distribution of entity types in an average benchmark assessment. From the ordered sample, we sequentially select the focal entity for the question and extract its one-hop neighbours to form a subgraph. If no valid entity types are present in a sample, a new one is generated.

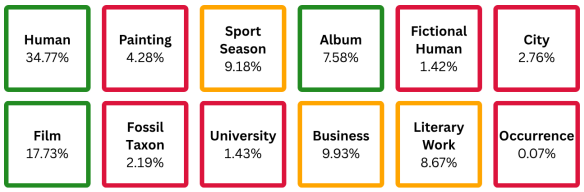

Figure 2: Distribution of Very Common, Common, and Uncommon Entity Types within the KGHaluBench assessment

3.1.2 Attributes Retrieval:

We retrieve three key sets of information from the focal entity's sub-graph and external database for the downstream process in KGHaluBench.

KG Triples are required both for generating the benchmark questions and verifying the correctness of the LLM's response. From the entity's subgraph, we extract pairs of facts and connecting relations and filter them against a predefined set of valid relations. As non-textual (e.g., *image*), trivial (e.g., *given name*), and irrelevant (e.g., *official website*) relations are unsuitable for generating challenging and coherent questions. From the remaining pairs, we randomly select three to form the question and validate the response. If fewer than three valid pairs exist, we lack the number to formulate the question. Therefore, we discard the current target entity and select the next focal entity from the KG sample.

Entity Description is a comprehensive overview of the entity from an external database. utilise the description as a factual representation when comparing against the LLM's response in the entity-level filter, as described in Section 3.2.

Entity Statistics are used within the question difficulty calculation to estimate the popularity of the entity, as detailed in Section 3.1.3. Similarly, in the KG-triple filtering process, if any statistic is null or invalid, we cannot quantify the entity's popularity; therefore, we must select the next focal entity from the sample set.

3.1.3 Entity Popularity:

Well-known entities are more likely to be referenced in an LLM's training data, increasing the likelihood that the LLM will accurately recall information about them. Therefore, estimating entity popularity provides valuable insight into the challenge posed by the question. To determine an entity's popularity, we combine the entity's individual relevance and the relevance of its associated type.

Entity Relevance: An entity's relevance is determined by two key factors: its **prominence**, reflecting recognisability and graph connectivity, and its **information coverage**, capturing the availability and detail of the information. To estimate relevance, we aggregate the following statistics from the KG and external databases.

- **Page Views**: Number of entity page views (2017–2025)
- **Site Links**: Number of site links from other entities
- **Linked Entities**: Number of neighbours in the KG
- **IDs**: Number of external database identifiers
- **Wiki Count**: Token count of the entity's page
- **Statements**: Number of relation–fact pairs
- **References**: Number of sources validating the KG facts

However, some statistics provide a stronger indication of the LLM's ability to answer questions accurately. To account for this, we derive an associated weight for each statistic using a machine learning pipeline as outlined in Appendix A.2.3.

Entity Type Relevance: In the LLM's training data, some entity types (e.g., *fictional characters*) frequently appear in descriptive texts, leading to a more comprehensive representation. In contrast, other entities (e.g. *paintings*) are usually referenced in shorter, literal descriptions and are less likely to be retained. To quantify this difference, we calculate entity-type weights by averaging the question scores for each entity type and normalising them to the range [0, 1].

3.1.4 Question Formation:

A key feature of KGHaluBench is the depth of its questions, which challenge the LLM's knowledge. We achieve this utilising the KG's relational structure to formulate compound questions about a single entity. As shown in Figure 3, our question template prompts the LLM to provide a brief overview of the entity and specific facts. This format is intentionally aligned with the entity's description to facilitate an accurate comparison during response verification.

The template requires details such as the entity's name and three randomly chosen valid relations. We provide supplementary context about the entity to reduce ambiguity when multiple entities may share the same name, enabling the LLM to give a more accurate response without revealing any examined information. The completed question template is supplied to LLM, and the response is collected for verification. Implementation details are provided in Appendix A.2.1.

Question Template - Example

Please provide a brief overview of **Robin Williams** (**1951–2014**). Your response should be around 200 words long and must include key details such as **occupation**, **place of birth**, and **children**. Focus exclusively on the most relevant entity associated with the provided name.

Figure 3: Question Template used for Question Generation.

Question Complexity: To correctly answer each relation in the question, the LLM must recall specific facts; however, the availability of such knowledge can vary considerably. For instance, the album's artist is often better known and more widely documented than the record label that produced it.

To account for this, we adopt the same weight calculation process as used in Section 3.1.3. As each entity type is associated with a distinct set of relations, we treat relation sets as independent. Within each set, we calculate the average question score for each valid relation and apply min-max normalisation to obtain relation weights. Question complexity is then estimated by aggregating the weights associated with the three relations it contains, reflecting the challenge the assessment content poses to the LLM. However, when the response is judged to be conceptually hallucinated, the complexity stems from the focal entity of the question rather than the relations. In this case, we approximate complexity by averaging the focal entity's relation set weights and multiplying the result by 3 to reflect the number of relations per question.

3.1.5 Question Difficulty:

Dynamic question generation introduces randomness, leading to variations in difficulty across assessments. Therefore, we incorporate a difficulty-scaled accuracy metric to ensure a fair and consistent benchmark.

We create a unified measure of question difficulty, Q_{d}, by aggregating entity popularity and question complexity. We drew inspiration from Item Response Theory, a psychometric paradigm for test scoring. The theory employs a sigmoid function, which, due to its S-shaped curve, allows for greater differentiation in the middle and the most common range of difficulties. This ensures that moderate-difficulty questions contribute equally to those at the extremes, resulting in a fairer representation of all question difficulties. Our modified sigmoid formula is illustrated below, where α controls the steepness of the sigmoid curve, Q is the question complexity, and Q_{Avg} represents the average question complexity across the three relations. EP denotes the entity popularity, which when min-max normalised is EP_{Norm}

$$Q_{\text{d}} = \frac{1}{1 + e^{-\alpha\left(Q_{\text{Avg}} - EP_{\text{Norm}}\right)}} \tag{1}$$

3.2 Response Verification

The *Response Verification Module* addresses the challenge of evaluating long-form responses by accurately identifying abstentions and potential hallucinations, then verifying factual correctness. Each component of the framework was validated against human judgment to ensure accuracy and effectiveness, as detailed in Appendix A.3.1.

3.2.1 Entity-Level Filter:

The entity-level filter classifies each response as aligned, hallucinated, or abstained, by identifying abstentions and evaluating the semantic and token-level similarities against the entity's description. Only aligned responses proceed to fact-level verification; hallucinated and abstained responses are scored accordingly and initiate another repetition of the question generation process.

We employ a small, efficient LLM to determine whether responses are abstentions or meaningful attempts to answer the question. The abstentions are defined as responses that refuse to answer, deflect the question, or admit to not recognising the focal entity. The models receive partial credit for appropriately expressing uncertainty or lack of knowledge (Kalai et al., 2025), as we assign one point to abstained responses.

For responses that meaningfully address the question, we evaluate their alignment with the entity using both conceptual overlap and token-level intersection. For the semantic comparison, we encoded both the response and the description, then quantified their similarity using cosine similarity. For token-level comparison, we use the intersection of common words between the LLM's response

and the entity's description. Entity-level similarity combines these metrics with a 70:30 ratio, prioritising semantic over lexical alignment, as the entity-level filter focuses on filtering out responses that misalign with the entity's domain. Responses exceeding the predefined threshold are considered aligned to the focal entity, while those below are deemed entity-level hallucinations and receive zero points. See Appendices A.2.2 and A.3.1 for further implementation details of the entity-level filter.

3.2.2 Fact-Level Check:

We implement a fact verification pipeline, as illustrated in Figure 4, to evaluate whether the relations specified in a question are correctly expressed in the LLM's response. Each relation is assessed independently as correct or incorrect, totalling a maximum of 3 points per response.

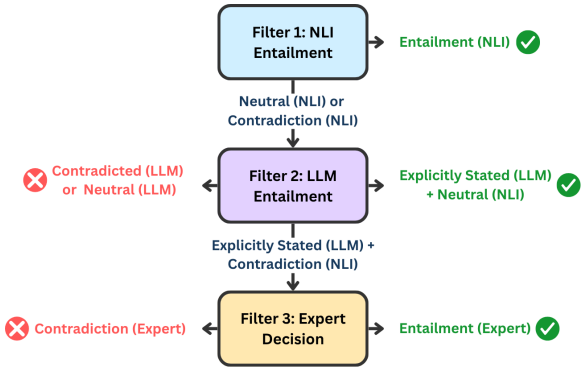

Figure 4: Overview of the fact verification pipeline

The pipeline requires an interpretable fact to compare to the LLM's response. Therefore, we employ an efficient LLM to transform the structured tuple of the entity's name, type, relation, tense indicator and fact into a sentence. The tense indicator is an auxiliary verb (e.g. 'is' or 'was') to ensure the constructed sentence is grammatically correct.

NLI Entailment Filter: For this filter, we utilise a Natural Language Inference (NLI) model. We chose this model because it is fast and efficient, while still achieving robust entailment recognition. We provided the NLI model with the reformatted fact and the LLM's response, requiring it to process the information and classify the outcome as entailment, contradiction, or neutral. If the result is entailment, we consider the fact correctly expressed in the LLM's response; otherwise, the response is passed to the LLM entailment filter.

LLM Entailment Filter: We employ an LLM in place of an entailment model, exploiting its advanced reasoning abilities to make the filter more accurate; however, it is more computationally expensive. We prompt the LLM to assume the role of a fact-checking assistant, tasked with determining whether the following fact is explicitly stated, contradicted, or not mentioned in the provided response. If a fact is stated incorrectly or omitted from the response, it is considered incorrect. Whereas, if the LLM concludes that the fact is explicitly stated, its decision is cross-validated against the NLI model. If the NLI result is neutral, we accept the LLM's judgment and consider the fact correctly expressed in the response. However, if the NLI model contradicts the LLM's decision, we proceed to the expert decision filter.

Expert Decision Filter: This filter functions as the final decision-maker. Within the pipeline, it is rarely used but serves as a fail-safe mechanism. We present the formatted fact alongside the response and create an ultimatum, with one expert's choice as entailment and the others as contradiction. We utilise the LLM to make the binary decision of which expert to agree with. If the LLM agrees with entailment, the fact is included in its response; otherwise, it is not.

4 Experiments

4.1 Knowledge Graph

Although our framework supports integrating any KG, we selected Wikidata for the experiments. This expansive knowledge base, roughly 118 million entities as of August 2025 (Wikidata, 2025), enables the curation of novel, challenging benchmarks to assess the LLM's knowledge.

4.2 Evaluation Models

We assessed 25 state-of-the-art LLMs on the KGHaluBench benchmark, as detailed in Appendix A.1. This included 15 open-source models ranging from 8 billion to 1 trillion parameters, and 10 proprietary models from 4 leading AI companies: OpenAI, Google, Anthropic, and xAI. Each model was evaluated using default parameters across 10 runs of 150 questions, with results aggregated using the mean. Open-source models run on Nebius AI Studio API, while proprietary models were accessed via their respective APIs.

4.3 Metrics

4.3.1 Weighted Accuracy:

We calculate *Accuracy* as the percentage of possible points earned through correct facts and ab-

stentions, rewarding models for both factuality and expressing uncertainty. However, to ensure we create a fair and consistent metric, $Accuracy$ must be scaled by the degree to which Q_{d} deviates from the average difficulty across all assessments, $Avg(Q_{\text{d}})$. We denote the new metric as W_{a}.

$$W_{\text{a}} = Accuracy \cdot \frac{Q_{\text{d}}}{Avg(Q_{\text{d}})} \tag{2}$$

4.3.2 Hallucination Rate:

We split the hallucination rate into breadth and depth of knowledge to provide more interpretability to an often one-dimensional metric.

Halu_{BOK} is the percentage of responses classified as hallucinations by the entity-level filter, which assesses the surface-level truthfulness between the LLM's response and the entity's description. A high rate indicates that the LLM lacks fundamental knowledge of the assessment content, suggesting limited breadth of knowledge.

$$\text{Halu}_{\text{BOK}} = \frac{|\text{Entity Hallucinations}|}{|\text{Total Responses}| - |\text{Abstentions}|} \tag{3}$$

Halu_{DOK} is the percentage of incorrect facts judged by the fact-level check. As only the responses that pass the entity-level filter are verified by the fact-level check, we divide the number of incorrect facts by the maximum possible score the model could achieve. A high rate indicates that while the LLM possesses the fundamental knowledge of the entity, it lacks sufficient depth to accurately answer the specified questions, suggesting a limited depth of knowledge.

$$\text{Halu}_{\text{DOK}} = \frac{|\text{Incorrect Facts}|}{\text{Maximum Attainable Score}} \tag{4}$$

5 Results

5.1 Weighted Accuracy

KGHaluBench presents a substantial challenge for the LLMs, as displayed in Figure 5, with *GPT-5* achieving the highest W_{a} of 65.60%. This difficulty arises from the question-generation process, as the random focal entity around which the compound question is constructed may reside anywhere from the centre to the periphery of the LLM's knowledge. This redundant performance gap provides longevity for the KGHaluBench, enabling it to maintain its relevance against the continually advancing knowledge capabilities of LLMs.

The challenge of KGHaluBench makes it an effective benchmark for contrasting the factual performance between different LLMs. To enable a fair comparison between open-source and proprietary technologies, we compute the average W_{a} for open-source models with $\geq$ 235B parameters and compare it against the average performance of all proprietary models. The proprietary models demonstrate superior factuality, achieving an average W_{a} of 55.94%, compared to 48.32%. This trend is evident: the five highest-performing models are all proprietary and show a clear increase in factuality compared to the rest. However, *GLM-4.5* achieves 54.35% W_{a}, outperforming powerful proprietary models, such as *Claude-4-Opus* and *Gemini-2.5-Flash*. This result indicates a narrowing performance gap between leading open-source and proprietary LLMs.

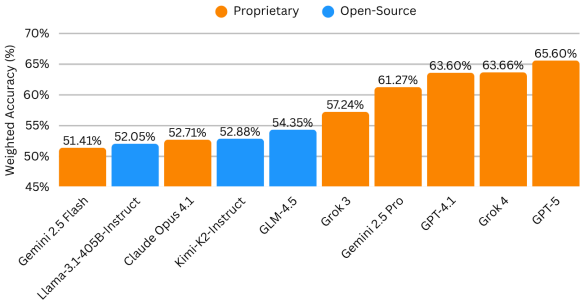

Figure 5: Top 10 Models by Weighted Accuracy. See Appendix A.1 for the full table of results.

5.2 Hallucination Rate

Decomposing the hallucination rate provides valuable insight into the factors that contribute to hallucinations across different-sized models. To quantify these trends, we compare averages from the smaller 8-32B models with those from larger proprietary models. As shown in Figure 6, the

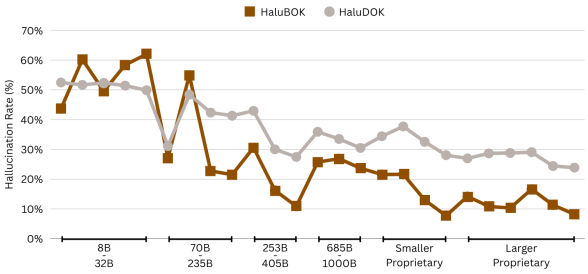

Figure 6: Halu_{BOK} and Halu_{DOK} hallucination rates across all models

Halu_{BOK} metric dramatically decreases, from 54.75% to 11.91%, as we move towards the larger, more proficient models. This trend suggests that the smaller models lack the competence to understand and recognise the focal entity in question. Therefore, the hallucinated responses are con-

structed from knowledge which is loosely or even unrelated to the entity.

In contrast, the Halu_{DOK} metric exhibits a more modest decrease by approximately 24.59% from the 8-32B models to the larger proprietary models. This difference between the two metrics widens as the models become more proficient, suggesting that larger models excel at determining what to discuss but still struggle to retain the precise details of that topic. Therefore, the hallucinations are not due to a failure in entity recognition, but instead to a lack of deeper or more specific knowledge.

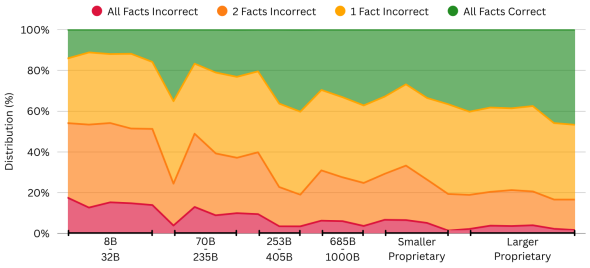

Figure 7: Distribution of fact-level hallucinations across all models, showing the proportion of responses containing 0, 1, 2, or 3 hallucinated facts

To further investigate this trend, we examine the fact-level hallucinations distribution across the models, as illustrated in Figure 7. The results show that more proficient models generate a lower proportion of highly detrimental responses, as those containing two or three hallucinated facts decrease from 52.91% to 19.09%. In turn, the percentage of responses that are entirely factually correct increases by around 28.18%. However, roughly 39.74% of responses from the more proficient models contain a single hallucinated fact. This undermines trustworthiness and creates a false sense of reliability, as the responses which appear fluent and factual still often include hallucinations.

5.3 Abstention Rate

Abstention is an expression of uncertainty, where a model refrains from answering the question if it lacks the understanding or knowledge to provide an accurate response. We integrate abstention detection in our entity-level filter, detailed in Section 3.2.1, enabling the classification of each response as aligned, hallucinated, or abstained. Figure 8 reveals that among models with comparable alignment rates, those with higher abstention show lower hallucination rates. For example, *Llama-3.1-8B* and *Gemma-2-9B* both achieve a 38.80% alignment between their responses and the focal entities' descriptions, indicating an equivalent breadth of knowledge. However, *Llama-3.1-8B* has a higher abstention rate, resulting in a significantly lower hallucination rate of 30.00%, compared to *Gemma-2-9B* of 58.67%. While this mechanism clearly reduces hallucinations, over-abstaining can severely limit the usefulness of the model. Several models demonstrate this problem with *GPT-oss-20B*, *GPT-5-Mini*, and *Claude Sonnet 4*, declining to answer 55.93%, 49.67%, and 49.40% of the benchmark questions, respectively. Therefore, we believe models should implement constructive abstention, in which the response provides guidance on how users can locate reliable information themselves.

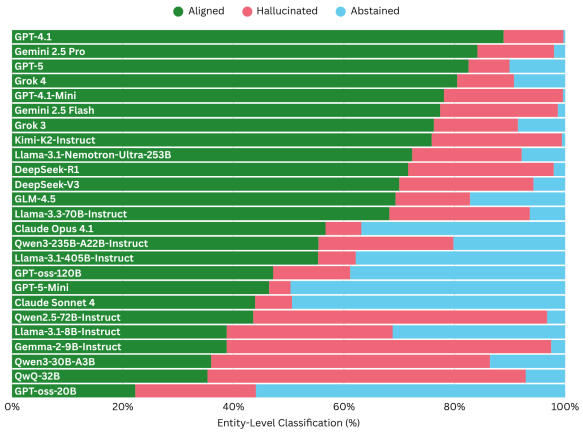

Figure 8: Distribution of entity-level filter classifications across all models, showing the proportion of responses classified as aligned, hallucinated, or abstained

6 Conclusion

We introduce KGHaluBench, a comprehensive hallucination benchmark that evaluates LLMs with questions that probe their depth and breadth of knowledge. Customisable framework dynamically generates multifaceted compound questions from randomly selected entities in the KG. These questions are then assessed at a conceptual level, and if no hallucinations or abstentions are detected, they are verified for factual correctness. Our experiments assess 27 state-of-the-art LLMs, utilising our novel metrics. We propose W_{a}, a fairer accuracy metric which adjusts standard accuracy by statistically estimating question difficulty. We also present Halu_{BOK} and Halu_{DOK}, as more interpretable hallucination metrics to provide deeper insights into factors that lead different-sized LLMs to hallucinate. We hope KGHaluBench can be used to develop effective strategies in addressing the challenge of mitigating hallucinations.

7 Limitations

In our benchmark, we rely on Wikidata as the information source, limiting the set of fact-seeking questions to entities within the KG. This introduces data quality limitations, as Wikidata's representation of entities is uneven across topics, cultures, and languages, particularly under-representing non-English entities and marginalised topics. Therefore, we would like to expand the benchmark further to include domain-specific KGs to explore hallucinations in specific scenarios. Additionally, our automated fact verification framework will make mistakes, potentially rejecting valid responses or scoring misaligned ones. Therefore, despite achieving substantial agreement with human judgment, as displayed in A.3.1, we must make improvements in accuracy and soundness of reasoning, which would further strengthen the validity and reliability of KGHaluBench.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, and 1 others. 2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg. Springer-Verlag.

Yejin Bang, Ziwei Ji, Alan Schelten, and 1 others. 2025. HalluLens: LLM hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24128–24156, Vienna, Austria. Association for Computational Linguistics.

Jonathan Berant, Andrew K. Chou, Roy Frostig, and 1 others. 2013. Semantic parsing on freebase from question-answer pairs. In *Conference on Empirical Methods in Natural Language Processing*.

Akhiad Bercovich, Itay Levy, Izik Golan, and 1 others. 2025. Llama-nemotron: Efficient reasoning models. *Preprint*, arXiv:2505.00949.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and 1 others. 2015. Large-scale simple question answering with memory networks. *Preprint*, arXiv:1506.02075.

Felix Busch, Lena Hoffmann, Christopher Rueger, and 1 others. 2025. Current applications and challenges in large language models for patient care: a systematic review. *Communications Medicine*, 5.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, and Ice Pasupat. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Preetam Prabhu Srikar Dammu, Himanshu Naidu, and Chirag Shah. 2025. Dynamic-kgqa: A scalable framework for generating adaptive question answering datasets. *Preprint*, arXiv:2503.05049.

DeepSeek-AI, Daya Guo, Dejian Yang, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, and 1 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, and 1 others. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 318–323, Minneapolis, Minnesota. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and 1 others. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *Preprint*, arXiv:2307.10169.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *Preprint*, arXiv:2509.04664.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, and 1 others. 2024. Realtime qa: What's the answer right now? *Preprint*, arXiv:2207.13332.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and 1 others. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, and 1 others. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Junyi Li, Xiaoxue Cheng, Xin Zhao, and 1 others. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2024. Large language models in finance: A survey. *Preprint*, arXiv:2311.10723.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Xiaoze Liu, Feijie Wu, Tianyang Xu, and 1 others. 2024. Evaluating the factuality of large language models using large-scale knowledge graphs. *Preprint*, arXiv:2404.00942.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, and 1 others. 2023. 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*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large language models: A survey. *Preprint*, arXiv:2402.06196.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and 1 others. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Pranav Narayanan Venkit, Tatiana Chakravorti, Vipul Gupta, and 1 others. 2024. An audit on the perspectives and challenges of hallucinations in NLP. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6528–6548, Miami, Florida, USA. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Qwen, An Yang, Baosong Yang, and 1 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

A B M Ashikur Rahman, Saeed Anwar, Muhammad Usman, and Ajmal Mian. 2024. Defan: Definitive answer dataset for llms hallucination evaluation. *Preprint*, arXiv:2406.09155.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, and 1 others. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *Preprint*, arXiv:2409.10173.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, and 1 others. 2024. Head-to-tail: How knowledgeable are large language models (llms)? a.k.a. will llms replace knowledge graphs? *Preprint*, arXiv:2308.10168.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

5 Team. 2025a. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *Preprint*, arXiv:2508.06471.

Gemma Team, Morgane Riviere, Shreya Pathak, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Kimi Team. 2025b. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and 1 others. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Tu Vu, Mohit Iyyer, Xuezhi Wang, and 1 others. 2023. Freshllms: Refreshing large language models with search engine augmentation. *Preprint*, arXiv:2310.03214.

Jason Wei, Nguyen Karina, Hyung Won Chung, and 1 others. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Wikidata. 2025. Wikidata:statistics - wikidata.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *ArXiv*, abs/2401.11817.

An Yang, Anfeng Li, Baosong Yang, and 1 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, and 1 others. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, and 1 others. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

Yanxu Zhu, Jinlin Xiao, Yuhang Wang, and 1 others. 2024. Kg-fpq: Evaluating factuality hallucination in llms with knowledge graph-based false premise questions. *Preprint*, arXiv:2407.05868.

A Appendix

A.1 Experimental Results

Table 1 details the weighted accuracy, abstain rate, and both hallucination rates for all models tested within the KGHaluBench experiments.

A.2 Detailed Methodology

A.2.1 Supplementary Context in the Question Template

The supplementary context provided in the question varies depending on the type of the focal entity. If the entity type is a human, the supplementary context is their lifespan: (1934-2023) if deceased, or their birth year to the present (1968-). For all other types, the context would be the entity type.

A.2.2 Detailed Entity-Level Filter

For the semantic comparison, we embedded both the response and the description into 1024-dimensional vectors using Jina-Embedding-V3 (Sturua et al., 2024). The model achieved a Sentence Textual Similarity (STS) score of 85.80 on the MTEB benchmark while maintaining an efficient parameter count of 570 million, making it an ideal choice for entity-level filtering (Muennighoff et al., 2023). We then quantified the similarity between the two texts using cosine similarity.

Model Version	Knowledge Cut-off	Model Size	W_a	Abstain Rate	Halu_{BOK}	Halu_{DOK}
Open-Source Models	(Date)	(Billion)	(%)	(%)	(%)	(%)
Meta-Llama-3.1-8B-Instruct-fast	12/2023	8	28.89	31.20	43.64	52.46
gemma-2-9b-it-fast	-	9	19.54	2.53	60.21	51.67
GPT-oss-20b	06/2024	20	29.45	55.93	49.56	52.32
Qwen3-30B-A3B	-	30	22.20	13.60	58.36	51.41
QwQ-32B-fast	11/2024	32	20.18	7.13	61.99	49.88
Llama-3.3-70B-Instruct-fast	12/2023	70	49.04	6.40	27.13	31.10
Qwen2.5-72B-Instruct	12/2023	72	23.67	3.27	54.96	48.39
GPT-oss-120b	06/2024	120	40.62	38.87	22.74	42.36
Qwen3-235B-A22B-Instruct-2507	-	235	44.94	7.87	21.49	41.32
Llama-3_1-Nemotron-Ultra-253B-v1	12/2023	253	38.61	20.20	30.57	42.95
GLM-4.5	-	335	**54.35**	17.20	<u>16.24</u>	<u>30.01</u>
Meta-Llama-3.1-405B-Instruct	12/2023	405	52.06	37.87	**10.88**	**27.46**
DeepSeek-V3-0324	07/2024	685	47.00	5.73	25.72	35.90
DeepSeek-R1-0528	01/2025	685	48.45	2.07	26.87	33.50
Kimi-K2-Instruct	-	1000	<u>52.88</u>	0.60	23.66	30.47
Proprietary Models						
GPT-4.1-mini-2025-04-14	06/2024	-	48.13	0.40	21.62	37.71
GPT-4.1-2025-04-14	06/2024	-	63.60	0.33	10.84	28.65
GPT-5-mini-2025-08-7	06/2024	-	49.58	49.67	**7.69**	28.05
GPT-5-2025-08-07	10/2024	-	**65.60**	10.07	<u>8.20</u>	**23.87**
Gemini-2.5-flash	01/2025	-	51.41	1.33	21.55	34.41
Gemini-2.5-pro	01/2025	-	61.27	2.00	14.18	26.97
Claude-sonnet-4-20250514	03/2025	-	46.18	49.40	13.01	32.55
Claude-opus-4-1-20250805	03/2025	-	52.71	36.87	10.32	28.79
Grok-3	11/2024	-	57.24	8.53	15.20	29.05
Grok-4-0709	11/2024	-	<u>63.66</u>	9.27	10.27	<u>24.40</u>

Table 1: Performance of 25 language models on the KGHaluBench using the metrics of W_a, Halu_{BOK}, and Halu_{DOK}, along with their version specifications. All models were evaluated across 10 assessments, each comprising 150 questions, with results aggregated using the mean. Models were evaluated using default parameters in accordance with their respective usage policies. The best performing results for the open-source and proprietary models are in **bold**, with the second-best values <u>underlined</u>. The knowledge cut-off dates were collected from https://github.com/HaoooWang/llm-knowledge-cutoff-dates. Citations: Llama families (Grattafiori et al., 2024; Bercovich et al., 2025), Qwen families (Qwen et al., 2025; Yang et al., 2025), Gemma2(Team et al., 2024), GLM-4.5 (Team, 2025a), Kimi-K2 (Team, 2025b) DeepSeek families (DeepSeek-AI et al., 2025a,b), GPT families (OpenAI, 2024), Gemini families (Comanici et al., 2025)

$$\mathbf{w} = \text{Embed(Description)}, \quad \mathbf{r} = \text{Embed(Response)}$$

$$\text{SemanticSim} = \frac{\mathbf{w} \cdot \mathbf{r}}{\|\mathbf{w}\|\|\mathbf{r}\|} \quad (5)$$

We calculate the token-level comparison using the Fuzzy Set Ratio from the RapidFuzz Module, which is based on the Levenshtein Distance Formula (Levenshtein, 1965). The Fuzzy Set Ratio uses the intersection of common words between the LLM's response and the entity's description, which is optimal for texts which differ in order and length, as is common in this situation. The Fuzzy Set Ratio, like cosine similarity, produces a similarity score represented as a percentage.

$$TokenSim = \left(1 - \frac{D(\text{Description},\text{Response})}{\max(|\text{Description}|,|\text{Response}|)}\right) \quad (6)$$

We aggregate entity similarity with a bias toward semantic meaning, as it better captures the conceptual relationship between the two texts.

$$EntitySim = 0.7 \cdot SemanticSim + 0.3 \cdot TokenSim \quad (7)$$

A.2.3 Weights Configuration

We created a calibration dataset by evaluating 34 selected models (1B-685B parameters, including both open-source and proprietary models) across 10 runs of 150 questions, resulting in a total of 51,000 data points. For each question, the dataset captures the corresponding entity's ID, statistics, and type, as well as the question's relation types, their scores, and the overall question score.

To ensure a balanced representation of all models in the weight configuration, we used stratified sampling to split the experiment dataset into training and validation sets at a 70:30 ratio. For each model's 1,500-question section, 70% (1,050 questions) were randomly selected for the training set, and the remaining 30% (450 questions) were reserved for validation.

To estimate the entity relevance weights, we employed TPOT to automatically generate a machine learning pipeline optimised to predict how well-known an entity is to the LLM, based on the entity's statistics. Before the training, we normalised each statistic using a natural logarithm transformation to ensure consistent scaling across features. The final pipeline includes max-abs feature scaling, feature selection using an ExtraTreesClassifier, parallel feature transformations via FeatureUnion, and a final ensemble using a BaggingClassifier. All hyperparameters and component choices were selected using TPOT's approach, which balances accuracy and model complexity. Finally, the learned feature coefficients were extracted and interpreted as entity-relevance weights.

Entity type relevance weights and relation complexity weights were also derived from the training data, following the processes outlined in Section 3.1.3 and Section 3.1.4, respectively.

A.3 Ablation Studies

We conduct an ablation study to isolate the contributions of the response verification framework and the weighted accuracy calculation. We demonstrate that both stages of the verification process closely align with human judgment. Additionally, we demonstrate that incorporating difficulty-based scaling into the accuracy metric results in a more consistent and fair assessment.

A.3.1 Response Verification Framework

We conducted a human validation study to evaluate the accuracy of the verification framework in comparison to human judgment. We tasked nine participants with answering 900 questions: 495 emulating the role of the entity-level filter and 405 replicating the role of the fact-level checker. For the entity-level filtering task, participants compared the LLM's response with the entity's Wikipedia description to determine if they referred to the same entity, disregarding any fact-level hallucinations. For the fact-level filtering task, participants were given three facts corresponding to the relations in the question and had to verify whether each was explicitly stated in the LLM's response. We compared our framework against an automated judge which utilised *GPT-3.5-Turbo*, as often used in literature. The nine participants, all Master's or PhD students, provided consent and were made aware that their responses would be used to evaluate and validate the verification framework.

Entity-Level Filter: Table 2 illustrates how varying the threshold influences the balance between precision and recall in the entity-level filter. We chose the specific thresholds after observing several KGHaluBench assessments to estimate the boundary between factual and hallucinated responses. The highest threshold of 0.750 achieved the greatest alignment with the human judges, at 79.19%. However, the lower recall of 73.17% indicates that the filter is overly strict for initial conceptual overlap comparisons, resulting in valid responses being

wrongly rejected. Lowering the threshold increases recall faster than it decreases precision, resulting in the highest F1 score of 78.07% at a threshold of 0.700, despite the overall agreement dropping slightly to 77.98%. Since the entity-level filter is at the first stage of the pipeline, we prioritise recall, as misaligned responses admitted at this stage will score poorly at the fact-level check, whereas aligned responses mistakenly discarded are detrimental to the assessment's accuracy score. These results suggest that the entity-level filter effectively differentiated between conceptually hallucinated and non-hallucinated responses, in agreement with the participants' decisions.

Model		P %	R %	F1 %	A %
KGHaluBench	**0.750**	75.76	73.17	74.44	79.19
	0.725	70.45	84.88	76.99	78.99
	0.700	66.44	94.63	78.07	77.98
GPT-3.5-Turbo	–	82.46	45.85	58.93	73.54

Table 2: Entity-Level Filter Performance Across Thresholds Compared to *GPT-3.5-Turbo* (P = Precision, R = Recall, A = Human Agreement)

We compared the entity-level filter against an automated judge using a backbone of *GPT-3.5-Turbo*. Our 0.700 threshold achieved 5.65% higher alignment with human judgment and 48.78% higher recall. These results suggest that the entity-level filter is a far more effective approach than a typical automated judge, as it can reliably determine whether the focal entity is consistent across the two texts despite potential nuances and discrepancies.

Fact-Level Check: In the human validation study, our tri-stage fact verification pipeline achieved 87.74% alignment with human judgment, which was 8.56% higher than the automated judge at 79.18%. To conduct a further investigation into the pipeline, we utilise the experimental data to highlight the contribution and efficiency of each filter.

Filter	Facts Verified (%)	Avg Time (s)
NLI Entailment	27.58	0.36
LLM Entailment	71.39	1.91
Expert Decision	1.03	2.35
Average Verification Time per Fact (s): 1.49		
Average Verification Time per Question (s): 4.47		

Table 3: Distribution of verified facts and average processing time in the fact verification pipeline

The first filter of the pipeline employs *RoBERTa-Large-MNLI* as an efficient preliminary check. This stage contributes most to the pipeline's overall efficiency, resolving 27.58% of the facts with an average verification time of 0.36 seconds, as shown in Table 3. The model's effectiveness is due to its lightweight design, with just 365M parameters. However, *RoBERTa-Large-MNLI* has limited reasoning capabilities, which necessitates a second layer of verification. Therefore, we position the filter at the forefront of the pipeline, where it primarily encounters facts that are easily verifiable, such as explicitly stated or numerical ones.

The secondary check is managed by the LLM entailment filter, which utilises the semantic reasoning capabilities of *Llama3.1:8B*. This stage contributes most to the pipeline's overall accuracy, as it handles cases with subtle discrepancies between the response and the grounded fact, such as changes in word order, phrasing, or nuance. Although roughly five times slower than the first filter, it still resolves 71.39% of the facts, indicating its importance in maintaining the integrity of the fact verification pipeline.

The Expert Decision filter resolves only 1.03% of the facts that pass through the fact verification pipeline. This outcome is expected, as a fact must meet a particular set of conditions in the first two filters before reaching this final fail-safe stage. Overall, the fact verification pipeline is efficient and accurate, which contributes to the validity of the results from KGHaluBench.

A.3.2 Question Difficulty Weighting

We utilise the training section of our calibration dataset, as detailed in Appendix A.2.3, to configure the weights used in the question difficulty calculation. To validate this metric, we assess how well the estimated difficulty aligns with the LLM's experienced difficulty by using the validation portion of the dataset. We use the question score to represent the actual difficulty, with higher scores indicating more straightforward questions. We then compared actual difficulty with estimated question difficulty and evaluated the relationship using Spearman's and Kendall's rank correlation coefficients to provide robust ordinal association measures.

The results show a moderate negative correlation between actual and estimated question difficulty, with Spearman's ρ = -0.403 and Kendall's τ = -0.299. Kendall's τ is marginally lower, as it only considers the number of pairwise disagreements, whereas Spearman's ρ also accounts for the dis-

tance between the pairs. This confirms that our metric captures the difficulty of the content within the benchmark questions.

To explore the practical impact of this metric, we apply the derived weights to calculate question difficulty scores for our main experimental evaluation. We assess whether it can improve assessment fairness by addressing the randomness and entity-popularity bias introduced by the question-generation mechanism. For each of the 25 models, we compute the mean accuracy and the weighted accuracy (W_a), along with their standard deviations, across 10 runs. These values are then averaged across all models to obtain aggregated metrics.

Metric	Mean Acc. (%)	Std. Dev. (%)
Accuracy	45.30	2.57
W_a	45.25	2.45
	-0.05	-0.12

Table 4: Mean and Standard deviation of Accuracy and W_a, over the 10 runs per model. Then averaged across all models in the experiment dataset

As shown in Table 4, applying difficulty-based weighting reduces the mean standard deviation across models by 0.12%, from 2.57% to 2.45%. This improvement in consistency demonstrates that W_a mitigates entity-popularity bias across assessments, resulting in a fairer, more stable benchmark. Notably, the weighting has a negligible effect on the mean accuracy, decreasing it by 0.05% to 45.25%. This ensures that W_a continues to reflect the benchmark's intended difficulty, preserving the validity of the accuracy results.

A.4 Implementation Details

A.4.1 Benchmark Prompts

System Prompt for all LLMs

```
Answer questions about specific entities. If you recognise the focal entity, give a clear, detailed response covering all requested details. If you do not know it well enough, state that you cannot answer.
```

Benchmark Template for all LLMs

```
Please provide a brief overview of [Entity Name] [Extra Context]. Your response should be around 200 words long and must include key details such as [Entity Relation 1], [Entity Relation 2], and [Entity Relation 3]. Focus exclusively on the most relevant entity associated with the provided name.
```

Parameters: Given the model's default.

A.4.2 Response Verification Prompts

Entity-Level Prompts: Prompt used within the entity-level filter.

Abstention Filter (Llama3.2:3b)

```
You are given a question and a response from a language model. Determine if the response abstains from answering the question. A response counts as abstaining if it refuses, deflects, or avoids providing the requested information, or if it only gives generic disclaimers without including the required details. If the response meaningfully answers the question, classify it as 'Answered'.

Return only one word: 'Abstained' or 'Answered'.

### Question: [Benchmark Question]

### Response: [LLM Response]
```

Parameters: Temperature: 0, Top_P: 0.6

Fact-Level Prompts: Prompt used within the fact verification pipeline.

Tuple to Golden Fact Translation

```
Convert the following structured tuple into a natural language sentence, using the [Entity Name] as the subject: ([entity Name] ([Entity Type]) , [Entity Relation] , [Verb] , [Fact]). Return only the resulting sentence and do not reword [Entity Name] or [Fact].
```

Parameters: Temperature: 0.3, Top_P: 0.5

LLM Entailment Filter (Llama3.1:8b)

```
You are a fact-checking assistant. Your task is to determine whether the following fact is explicitly stated, supported, contradicted, or not mentioned in the provided response.

### Response: [LLM Response]

### Fact: [Golden Fact]

### Response Options: Respond with one of the following options and a brief explanation:

- EXPLICITLY STATED: The fact is directly and clearly stated in the response, using the same or equivalent wording. Numerical or time-related facts must match exactly. - CONTRADICTED: The fact is directly contradicted by information in the response. - NOT MENTIONED: The fact is not present in the response, and there is no sufficient evidence to confirm or contradict it.

Only return one of the four options and a single concise explanation. Do not provide additional commentary.
```

Parameters: Temperature: 0, Top_P: 0.6

Expert Entailment Filter (Mistral:7b)

```
You are a fact-checking assistant. Your task is to determine whether Expert 1 or Expert 2 is correct based on the provided response and fact.

### Response: [LLM Response]

### Fact: [Golden Fact]

Expert 1: Entailment (The response aligns with the fact.) Expert 2: Contradiction (The response contradicts the fact.)
```

```
Return only "Expert 1" or "Expert 2" based on the
correct evaluation. No explanations.
```

Parameters: Temperature: 0, Top_P: 0.6

A.4.3 Automated Judge Prompts

Prompts used in the human validation study for comparing KGHaluBench's entity-level and fact-level filters against using an automated judge of GPT-3.5-Turbo.

GPT Entity-Level Filter

```
You will be given two passages: - A response generated
by a large language model. - A reference description
of a target entity.

Your task is to determine if both passages refer to the
**same entity**.

Check both: 1. Semantic similarity (do they describe
the same person/place/thing, even with different
wording?) 2. Token-level match (e.g., name or
identifiers must match exactly or with minor variation)

Respond with only one word: True if they refer to the
same entity, or False if they do not.

### Response: [LLM Response]

### Description: [Entity Description]

Output:
```

Parameters: Temperature: 0, Max Tokens: 10

GPT Fact-Level Check

```
You will be given: - A golden fact (a specific claim or
statement) - A response generated by a language model

Your task is to determine whether the golden fact
is **explicitly stated or clearly implied** in the
response.

Only respond with: - True — if the fact is stated
or clearly implied - False — if the fact is missing,
contradicted, or too vague

### Golden Fact: [Golden Fact]

### LLM Response: [LLM Response]

Output:
```

Parameters: Temperature: 0, Max Tokens: 10