arXiv:2405.00648v1 [cs.SE] 1 May 2024

HalluVault: A Novel Logic Programming-aided Metamorphic Testing Framework for Detecting Fact-Conflicting Hallucinations in Large Language Models

NINGKE LI[*], Huazhong University of Science and Technology, China
YUEKANG LI[*], The University of New South Wales, Australia
YI LIU, Nanyang Technological University, Singapore
LING SHI, Nanyang Technological University, Singapore
KAILONG WANG[†], Huazhong University of Science and Technology, China
HAOYU WANG, Huazhong University of Science and Technology, China

Large language models (LLMs) have transformed the landscape of language processing, yet struggle with significant challenges in terms of security, privacy, and the generation of seemingly coherent but factually inaccurate outputs, commonly referred to as hallucinations. Among these challenges, one particularly pressing issue is Fact-Conflicting Hallucination (FCH), where LLMs generate content that directly contradicts established facts. Tackling FCH poses a formidable task due to two primary obstacles: Firstly, the hard task of automatically constructing and updating benchmark datasets, since current methods heavily rely on manually curated static benchmarks that fail to encompass the wide-ranging and evolving spectrum of FCH scenarios. Secondly, validating the reasoning process behind LLM outputs is inherently complex, especially when it involves intricate logical relations.

In addressing these obstacles, we propose an innovative approach leveraging logic programming to enhance metamorphic testing for detecting Fact-Conflicting Hallucinations (FCH). Our method introduces a sophisticated framework that leverages crawling techniques to gather information from sources such as Wikipedia, seamlessly integrated within our tool. By employing logical reasoning principles, we refine and expand this knowledge base to generate a diverse array of test cases with definitive answers. These test cases are then employed to assess LLMs through structured prompts, prompting them to deliver responses backed by logical reasoning. To validate the coherence of their reasoning, we present two semantic-aware assessment mechanisms that evaluate the alignment between the logical and semantic structures of the LLM responses and the ground truth. Our method autonomously generates pertinent test cases and detects hallucinations across six different LLMs spanning nine domains, revealing hallucination rates ranging from 24.7% to 59.8%. Key observations indicate that LLMs encounter challenges particularly with temporal concepts, handling out-of-distribution knowledge, and exhibiting deficiencies in logical reasoning capabilities. The outcomes underscore the efficacy of logic-based test cases generated by our tool in both triggering and identifying hallucinations. To further mitigate the identified FCHs, we delve into model editing techniques, which demonstrated effectiveness on a limited scale (involving edits to fewer than 1000 knowledge pieces). These findings underscore the imperative for ongoing collaborative endeavors within the community to detect and address LLM hallucinations.

1 INTRODUCTION

Large Language Models (LLMs) have revolutionized the landscape of language processing and beyond, exhibiting remarkable capabilities in text generation and comprehension with a wide range of applications. However, despite their growing prominence, LLMs encounter significant challenges in terms of security and privacy, which greatly impact their effectiveness and reliability. One particularly notable issue is the phenomenon of "hallucination", where LLMs

---

[*] Ningke Li and Yuekang Li are co-first authors.
[†] Kailong Wang is the corresponding author.

Authors' addresses: Ningke Li, Huazhong University of Science and Technology, China, lnk_01@hust.edu.cn; Yuekang Li, The University of New South Wales, Australia, yuekang.li@unsw.edu.au; Yi Liu, Nanyang Technological University, Singapore, yi009@e.ntu.edu.sg; Ling Shi, Nanyang Technological University, Singapore, ling.shi@ntu.edu.sg; Kailong Wang, Huazhong University of Science and Technology, China, wangkl@hust.edu.cn; Haoyu Wang, Huazhong University of Science and Technology, China, haoyuwang@hust.edu.cn.

generate coherent but factually incorrect or irrelevant outputs during tasks such as problem-solving. This tendency to produce misleading information not only compromises the safety of LLM applications but also raises serious concerns about their usability. Among the various forms of hallucinations in LLMs, "Fact-conflicting hallucination" (FCH) is a primary concern and the main focus of this paper. FCH occurs when LLMs generate content that directly contradicts established facts, as illustrated in Figure 2. For example, an LLM might incorrectly believe that "Haruki Murakami won the Nobel Prize in Literature in 2016", deviating from the correct answer that "Haruki Murakami has not won the Nobel Prize but has received numerous other awards for his work in Literature". Such misinformation dissemination leads to significant user confusion, eroding the trust and reliability that are essential in various LLM applications.

To tackle the issue of hallucinations in LLMs, recent studies have introduced various methods for their detection and testing. A common and straightforward approach involves creating extensive benchmarks tailored for this purpose. Datasets such as TruthfulQA, HaluEval, and KoLA have been designed to evaluate hallucinations across different contexts, including question-answering, summarization, and knowledge graphs. However, the current research landscape in LLM presents a critical gap in automatically and effectively testing FCHs. The main obstacle in testing for FCH is the absence of dedicated ground truth datasets and specific testing frameworks. Unlike other types of hallucinations (e.g., input-conflicting and context-conflicting hallucinations, to be detailed in Section 2), which can be identified through checks for semantic consistency, FCH requires the verification of the content's factual accuracy against external sources of knowledge or databases. This requirement makes the process particularly challenging and resource-intensive, especially for tasks processing contents with inherent logical connections.

Bridging the identified research gap in the literature necessitates an exploration of the inherent challenges faced in detecting FCHs, which are crucial for advancing and enhancing the reliability of LLMs. The first challenge is the difficulty in automatically constructing and updating benchmark datasets. Predominantly, existing methodologies rely on manually curated benchmarks. While these benchmarks are effective in detecting certain types of hallucinations, they fall short in encompassing the broad and dynamic spectrum of fact-conflicting scenarios inherent to LLMs. Moreover, the need for frequent updates to benchmark data, due to the ever-evolving nature of knowledge, imposes a significant and continuous maintenance effort. The reliance on benchmark datasets thus restricts the detection techniques' adaptability, scalability, and worse, detection capability. The second challenge is the difficulty in automatically validating answers from LLM outputs. Even when LLMs produce correct final answers, the outputs may not represent the true reasoning process behind them, potentially masking false understanding – a source of FCH hallucination. Automatically validating the reasoning process, especially those involving complex logic relations, is inherently difficult. Furthermore, the consistency in the quality of benchmark questions can vary due to the differing levels of experience and skill among human experts creating them, introducing noise, particularly in data labeling and result validation stages.

To address limitations in the existing techniques, we are the first, to the best of our knowledge, to introduce a novel automatic logic-programming-aided metamorphic testing technique for hallucination detection in this work. We have developed an extensive and extensible FCH testing framework, which is based on factual knowledge reasoning and metamorphic testing, seamlessly integrated into our tool called HalluVault. HalluVault begins by establishing a comprehensive factual knowledge base, sourced through extensive crawling of information from accessible knowledge bases such as Wikipedia. Each piece of this knowledge acts as a "seed" for subsequent transformations. Leveraging logic reasoning relations, we transform and augment these seeds, thereby expanding the factual knowledge into a well-established set of question-answer pairs. Using the questions and answers in the knowledge set as test cases and ground truth respectively, we construct a reliable and robust FCH testing benchmark. This is implemented through a series of well-formulated template-based prompts to test FCH in LLMs. Specifically, we instruct the LLMs to generate their answers to the test cases. To facilitate a thorough evaluation of the reasoning logic behind their responses, we require the LLMs to provide detailed justifications for their answers. For effective and dependable identification of FCH, we introduce two semantic-aware and similarity-based metamorphic oracles. These oracles operate by extracting essential semantic elements from each sentence and mapping out their logical relationships. By assessing the similarity between the constructed logical and semantic structures of the LLM's answers and the ground truth, we can detect FCH by pinpointing answers that significantly diverge from the ground truth.

In evaluating our proposed FCH testing framework and HalluVault, we undertake comprehensive experiments to evaluate their effectiveness in a wide array of contexts. On the one hand, our evaluation strategy involves deploying HalluVault across a broad spectrum of topics, sourced from an extensive and diverse range of Wikipedia articles. On the other hand, we examine our framework on a variety of open-source and commercial LLMs, providing a thorough examination of its applicability and performance across different model architectures.

Our key findings indicate that HalluVault succeeds in automatically generating useful test cases and identifying hallucination issues of six LLMs across nine domains. Using these test sets, we find that hallucination responses generated by different LLMs can vary from 24.7% to 59.8%. We then categorize these hallucination responses into four types. Through an in-depth analysis, we unveil that the lack of logical reasoning capabilities contributes the most to the FCH issues in LLMs. Additionally, we observe that LLMs are particularly prone to generating hallucinations in test cases involving temporal concepts and out-of-distribution knowledge. Furthermore, we confirm that test cases generated using our logical reasoning rules can effectively trigger and detect hallucination issues in LLMs. As mitigation, we investigate the use of model editing techniques to rectify the identified FCHs. These techniques have shown promising results when applied on a small scale (involving edits up to less than 1000 pieces of knowledge). Our results highlight the importance of ongoing efforts within the community to detect and address issues of hallucination in LLMs.

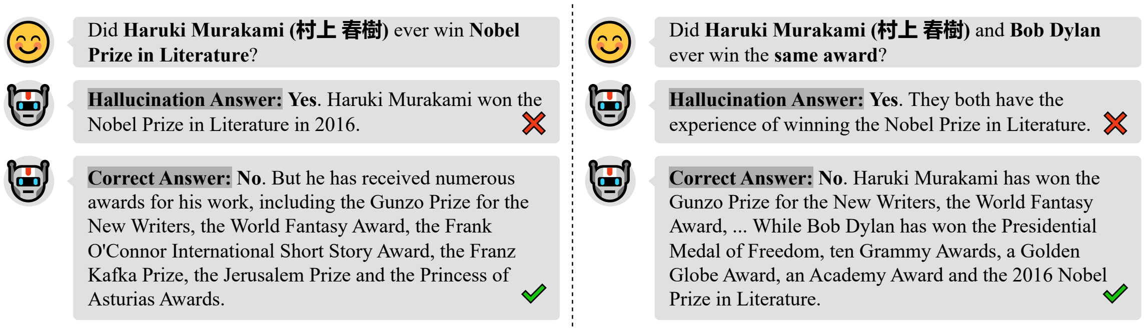
The main contributions of this paper are as follows:

- Development of a novel FCH Testing Framework: To the best of our knowledge, we are the first to develop a novel testing framework based on logic programming and metamorphic testing to automatically detect FCH issues in LLMs. This framework represents a significant advancement over current methodologies, providing a more systematic, comprehensive approach to detection.
- Construction and Release of Extensive Factual Knowledge Base and Benchmark: Our work constructs a large-scale benchmark dataset to facilitate collaborative efforts and future advancements in the detection of FCH.
- Designing and Implementing Innovative Logic-reasoning-based Method for Data Mutation: We propose and implement five unique logic reasoning rules to mutate and augment the initial seeds from our knowledge base, increasing the diversity and effectiveness of our test scenario.
- Deployment of FCH-specific semantic-aware testing oracles for automatic LLM answer validation: We propose and implement two automated verification mechanisms (oracles) that analyze the semantic structure similarity between sentences. These oracles are designed to validate the reasoning logic behind the answers generated by LLMs, hereby reliably detecting the occurrence of FCHs.

2 BACKGROUND

2.1 Hallucination Categorization

Hallucination in LLMs can be categorized into the three main categories [13, 46, 49], as detailed below.



Did **Haruki Murakami (村上 春樹)** ever win **Nobel Prize in Literature**?

Hallucination Answer: **Yes**. Haruki Murakami won the Nobel Prize in Literature in 2016. ✗

Correct Answer: **No**. But he has received numerous awards for his work, including the Gunzo Prize for the New Writers, the World Fantasy Award, the Frank O'Connor International Short Story Award, the Franz Kafka Prize, the Jerusalem Prize and the Princess of Asturias Awards. ✓

(a) Erroneous Knowledge Hallucination

Did **Haruki Murakami (村上 春樹)** and **Bob Dylan** ever win the **same award**?

Hallucination Answer: **Yes**. They both have the experience of winning the Nobel Prize in Literature. ✗

Correct Answer: **No**. Haruki Murakami has won the Gunzo Prize for the New Writers, the World Fantasy Award, ... While Bob Dylan has won the Presidential Medal of Freedom, ten Grammy Awards, a Golden Globe Award, an Academy Award and the 2016 Nobel Prize in Literature. ✓

(b) Erroneous Inference Hallucination

Fig. 1. A hallucination output example.

Input-Conflicting Hallucination: This type arises when LLMs produce outputs that are inconsistent with the user's input. This inconsistency can occur in two ways: either the model's response contradicts the task instructions (reflecting a misunderstanding of user intents) or the generated content contradicts the task input (similar to conventional issues in machine translation and summarization). An example of this would be an LLM replacing a key name or detail in a summary, deviating from the actual content provided by the user.

Context-Conflicting Hallucination: In this case, LLMs exhibit contradictions or inconsistencies in lengthy or multi-turn responses. This happens when models lose track of the context or fail to maintain consistency throughout the conversation. Limitations in maintaining long-term memory or identifying relevant context are often the culprits. An instance of context-conflicting hallucination could involve LLMs switching references between two different individuals in a conversation about a specific topic.

Fact-Conflicting Hallucination: This type of hallucination is the key focus of this paper. It occurs when LLMs generate information that is in direct conflict with established world knowledge. This can be due to various factors introduced at different stages of the LLM lifecycle. For example, as shown in Figure 1, an LLM might provide incorrect historical information in response to a user's query, misleading users who are less knowledgeable about the subject.

In this paper, our primary focus is on fact-conflicting hallucinations, a type of error that carries the potential for more serious consequences by misleading users.

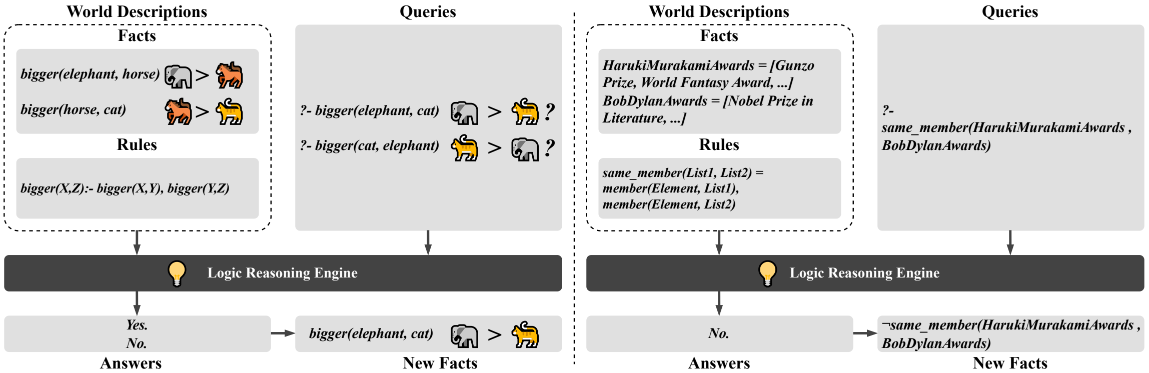
2.2 Logic Programming

Logic programming languages are declarative, i.e., programming with these languages means describing the world. Using the programs means asking questions about the previously described world. Based on the answers to the questions from the logic reasoning engine, according to the world description, we can acquire new facts. Figure 2 shows an example of how logic programming works. Here we explain some key concepts:

Facts. A fact is a statement defining a relation as being true. It is made up of a *predicate* and several *entities*. It is denoted as:

$$predicate(entity_1, entity_2, ...) \tag{1}$$

An example is $bigger(horse, cat)$, which means horses are bigger than cats. Another example is $member(GunzoPrize, HarukiMurakamiAwards)$, which means that Gunzon Prize is in the list of prizes awarded to Haruki Murakami.


World Descriptions
Facts
bigger(elephant, horse)
bigger(horse, cat)
Rules
bigger(X,Z):- bigger(X,Y), bigger(Y,Z)
Queries
?- bigger(elephant, cat)
?- bigger(cat, elephant)
Logic Reasoning Engine
Yes.
No.
Answers
bigger(elephant, cat)
New Facts
World Descriptions
Facts
HarukiMurakamiAwards = [Gunzo Prize, World Fantasy Award, ...]
BobDylanAwards = [Nobel Prize in Literature, ...]
Rules
same_member(List1, List2) = member(Element, List1), member(Element, List2)
Queries
?- same_member(HarukiMurakamiAwards , BobDylanAwards)
Logic Reasoning Engine
No.
Answers
¬same_member(HarukiMurakamiAwards , BobDylanAwards)
New Facts

(a) The animal size comparison example

(b) The prize comparison example

Fig. 2. Examples of logic programming.

Rules. A rule is a statement defining the logical relations of two or more facts and rules. It is made up of a *head* (a new predicate) and a *body* (a sequence of facts or rules separated by commas). It is denoted as:

$$predicate'(...) : -\ predicate_1(...), predicate_2(...), ... \tag{2}$$

An example is $bigger(X, Z) : -\ bigger(X, Y), bigger(Y, Z)$, which means the *bigger* relation is **transitive**. Another example is $smaller(X, Y) : -\ bigger(Y, X)$, which means *smaller* is an **inverse** relation of *bigger*. The last example here is $same\_member(List1, List2)$, which is *true* if there exists at least one *Element* that is a member of both $List1$ and $List2$. It is a **composition** of two *member* predicates.

Queries. A query has the same structure as the body of a rule, i.e., it is a sequence of predicates separated by commas. It is denoted as:

$$?-\ predicate_1(...), predicate_2(...), ... \tag{3}$$

The logic reasoning engine will answer Yes if the sequence of predicates in the query is $True$ according to the facts and rules. Otherwise, it will answer No. An example query is $?-\ bigger(elephant, cat)$, which means asking the logic reasoning engine whether elephants are bigger than cats. Another example is $?-\ same\_member(HarukiMurakamiAwards, BobDylanAwards)$, which means asking if the awards won by Haruki Murakami and Bob Dylan have overlaps.

Reasoning Rules. As shown in Figure 2, generating new facts through logic programming requires facts (Equation (1)), rules (Equation (2)), queries (Equation (3)), and answers to the queries. To simplify the notation of this process, we bring up the concept of *reasoning rules* in this paper, which describes the inference process of using facts and rules (predicates) to reach the conclusion (a new fact in the form of a predicate) by omitting the process of querying and analyzing the query answers. A reasoning rule is denoted in this form:

$$\frac{predicate_1(...), predicate_2(...), ...}{conclusion(...)} \tag{4}$$

3 MOTIVATING EXAMPLE

Figure 3 shows a motivating example of HalluVault. Assume we have the facts about whether Haruki Murakami and Bob Dylan have won the Nobel Prize, as illustrated in the left sub-figure. The question to ask LLMs is straightforward:

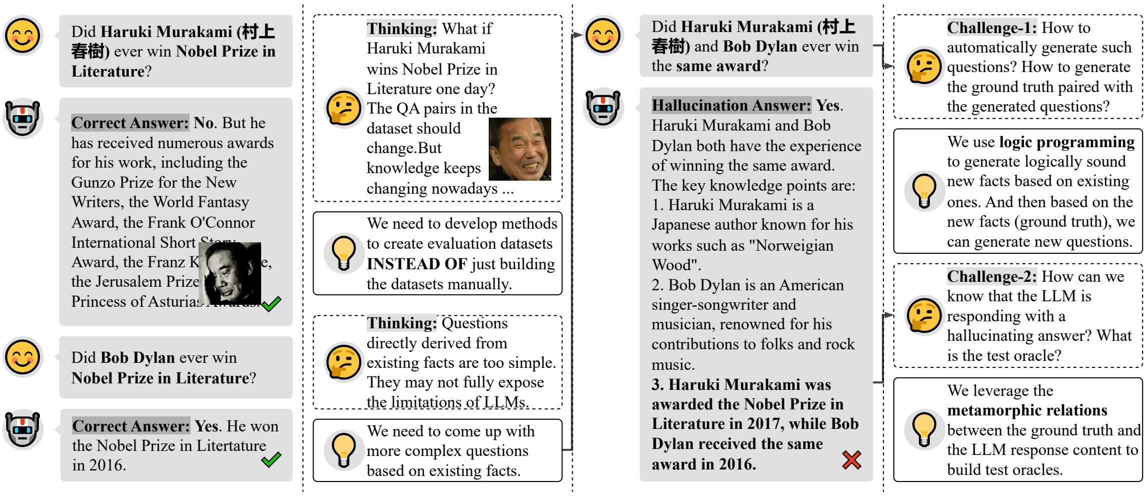

Did Haruki Murakami (村上春樹) ever win Nobel Prize in Literature?
Correct Answer: No. But he has received numerous awards for his work, including the Gunzo Prize for the New Writers, the World Fantasy Award, the Frank O'Connor International Short St Award, the Franz K e, the Jerusalem Prize Princess of Asturias
Did Bob Dylan ever win Nobel Prize in Literature?
Correct Answer: Yes. He won the Nobel Prize in Litertature in 2016.
Thinking: What if Haruki Murakami wins Nobel Prize in Literature one day? The QA pairs in the dataset should change.But knowledge keeps changing nowadays ...
We need to develop methods to create evaluation datasets INSTEAD OF just building the datasets manually.
Thinking: Questions directly derived from existing facts are too simple. They may not fully expose the limitations of LLMs.
We need to come up with more complex questions based on existing facts.
Did Haruki Murakami (村上春樹) and Bob Dylan ever win the same award?
Hallucination Answer: Yes. Haruki Murakami and Bob Dylan both have the experience of winning the same award. The key knowledge points are: 1. Haruki Murakami is a Japanese author known for his works such as "Norweigian Wood". 2. Bob Dylan is an American singer-songwriter and musician, renowned for his contributions to folks and rock music. 3. Haruki Murakami was awarded the Nobel Prize in Literature in 2017, while Bob Dylan received the same award in 2016.
Challenge-1: How to automatically generate such questions? How to generate the ground truth paired with the generated questions?
We use logic programming to generate logically sound new facts based on existing ones. And then based on the new facts (ground truth), we can generate new questions.
Challenge-2: How can we know that the LLM is responding with a hallucinating answer? What is the test oracle?
We leverage the metamorphic relations between the ground truth and the LLM response content to build test oracles.

Fig. 3. Motivating example.

We can ask whether Haruki Murakami/Bob Dylan has won the Nobel Prize or not. Asking and verifying this knowledge require no logic reasoning. However, the straightforward questions are often not enough to unveil hallucinations. **Therefore, more diversified questions (questions with intertwined and complex information, as illustrated in the right sub-figure) are needed.**

In order to generate more diversified benchmarks, previous research [17, 48] involves human experts to generate the questions and annotate the answers for hallucination checking. Although the manually generated benchmarks can unveil certain hallucinations, they suffer from several drawbacks. **The landscape of knowledge is dynamic, with new information continuously surfacing and older information becoming obsolete.** If facts change continuous over time, for instance Haruki Murakami were to win the Nobel Prize in the future, this would necessitate regular updates and corrections to the ground truth in existing datasets to reflect them. However, maintaining the accuracy of these benchmarks demands a significant amount of manual labor. Additionally, the quality of the questions in the benchmarks might be inconsistent due to the differences in the experience and skills of the human experts who create them. Consequently, the efficiency and soundness of the manually generated benchmarks does not have a guarantee.

The limitations of the manually generated benchmarks motivate the need for an automated technique to test for hallucinations in LLMs. Nevertheless, automatically generating diverse benchmarks is challenging. **First, generating suitable and valid questions is challenging (challenge#1).** While it is important for the questions in the testing benchmark to cover a diverse range of scenarios, they cannot be randomly generated or arbitrarily selected. Instead, the questions must be logically coherent and aligned with well-established factual knowledge and ground truth. **Second, deriving the test oracles for detecting hallucinations is challenging (challenge#2).** The LLM's answer is typically expressed in lengthy and potentially complex sentences. The key to determining if an LLM has produced an FCH lies in assessing whether the overall logical reasoning behind its answer is consistent with the established ground truth. Automatically analyzing and comparing the intricate logical structures within the LLM's response and the factual ground truth remains an inherently difficult task.

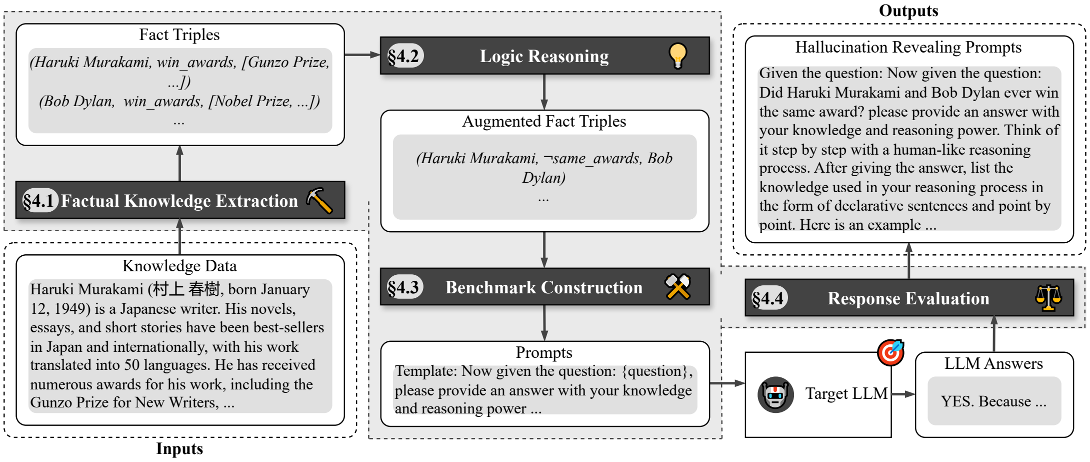

Outputs
Fact Triples
(Haruki Murakami, win_awards, [Gunzo Prize, ...])
(Bob Dylan, win_awards, [Nobel Prize, ...])
...
§4.2 Logic Reasoning
Augmented Fact Triples
(Haruki Murakami, ¬same_awards, Bob Dylan)
...
Hallucination Revealing Prompts
Given the question: Now given the question: Did Haruki Murakami and Bob Dylan ever win the same award? please provide an answer with your knowledge and reasoning power. Think of it step by step with a human-like reasoning process. After giving the answer, list the knowledge used in your reasoning process in the form of declarative sentences and point by point. Here is an example ...
§4.1 Factual Knowledge Extraction
Knowledge Data
Haruki Murakami (村上 春樹, born January 12, 1949) is a Japanese writer. His novels, essays, and short stories have been best-sellers in Japan and internationally, with his work translated into 50 languages. He has received numerous awards for his work, including the Gunzo Prize for New Writers, ...
§4.3 Benchmark Construction
§4.4 Response Evaluation
Prompts
Template: Now given the question: {question}, please provide an answer with your knowledge and reasoning power ...
Target LLM
LLM Answers
YES. Because ...
Inputs

Fig. 4. The workflow of HalluVault.

These two challenges can both be addressed by leveraging logic programming. We can derive new logically sound facts based on existing knowledge. With the new facts, we can then generate diverse questions and their ground truth answers. With the ground truth answers, we can generate test oracles to capture hallucinations. In short, the idea of using logic programming to tackle the challenges motivates the design of HalluVault.

4 METHODOLOGY

We design and implement HalluVault to address the aforementioned challenges, the workflow of which is illustrated in Figure 4. HalluVault is comprised of the following four modules, with each module to be detailed later.

- **Factual Knowledge Extraction (§4.1):** Based on voluminous knowledge database dumps, HalluVault acquires fundamental information and factual triples of valid entities.
- **Logical Reasoning (§4.2):** In this module, HalluVault leverages reasoning rules to generate sound and diverse facts as new ground truth knowledge.
- **Benchmark Construction (§4.3):** This module focuses on creating high-quality test case-oracle pairs from the newly-derived ground truth knowledge. The test oracles are generated based on a simple yet effective metamorphic relation: *Since the newly generated knowledge is sound, the questions complying with the knowledge should be answered with "YES" and the questions contravening the knowledge should be answered with "NO".* This module also includes strategies for effectively and reliably generating or selecting prompts for interaction with LLMs.
- **Response Evaluation (§4.4):** The final module evaluates the responses from the LLMs and detects factual consistency automatically. It first parses LLM outputs using NLP to construct semantic-aware structures, then evaluates their semantic similarity to ground truth. Subsequently, it develops similarity-based oracles applying metamorphic testing to assess consistency between LLM responses and ground truth.

Algorithm 1 Ground Truth Knowledge Triples Extraction

```
Require: Domain, SPARQL Endpoint, Entity Category, Relation Category
Ensure: Ground Truth Knowledge Triples $ground\_truth\_triples$
1: function ExtractGroundTruth($Domain, SPARQL\_Endpoint, Entity\_Category, Relation\_Category$)
2:     $entity\_list \leftarrow$ QueryEntities($Domain, SPARQL\_Endpoint, Entity\_Category$)    ▷ Query entities
3:     $relation\_list \leftarrow$ QueryRelations($Domain, SPARQL\_Endpoint, Relation\_Category$)    ▷ Query relations
4:     $ground\_truth\_triples \leftarrow [\,]$    ▷ Initialize ground truth knowledge triples
5:     for $entity$ in $entity\_list$ do    ▷ Iterate over each entity
6:         for $relation$ in $relation\_list$ do    ▷ Iterate over each relation
7:             $triples \leftarrow$ QueryTriples($entity, relation, Domain, SPARQL\_Endpoint$)    ▷ Query triples
8:             $ground\_truth\_triples$.append($triples$)    ▷ Add triples to ground truth knowledge base
9:     return $ground\_truth\_triples$    ▷ Return ground truth knowledge base
```

4.1 Factual Knowledge Extraction

This step aims to extract fundamental facts from the input knowledge data into fact triples that can be utilized for logical reasoning. A fact triple is denoted as:

$$(subject, relation, object) \tag{5}$$

where $subject$ and $object$ are entities in logic programming, and $relation$ is a predicate. In this paper, we use the notation in Equation (5) to denote the static data and use the notation in Equation (1) to denote the action of querying about the predicate. Given a piece of knowledge in natural language, the key is to identify the $subject$, $object$, and their $relation$.

Existing knowledge databases [4, 6, 26, 36] not only encompass a vast array of documents and pages but also provide available structured data. Extracted from knowledge databases, the structured data would become an ideal resource for the construction and enrichment of factual knowledge. Thus, the genesis of our test case data is exclusively rooted in the entities and structured information sourced from current knowledge databases, ensuring a sophisticated and well-informed foundation for our testing framework. Basically, we follow the categorization of entities and relations used by WikiPedia [4] to perform the identification. Figure 5a shows the categories of the entities. Figure 5b shows the categories of the relations and some example fact triples. The detailed process is outlined in Algorithm 1. For specific entities and relations from various domains (Lines 2-3), we automatically query the database for constructing the ground truth triples (Lines 6-8), providing useful information for the following components.

Category Type	Description
Culture and the Arts	Famous films, books, etc.
Geography and Places	Countries, cities and locations.
Health and Fitness	Diseases and disease-causing genes.
History and Events	Famous historical events, etc.
People and Self	Important figures and contributors.
Mathematics and Logic	Common formulas and theorems.
Natural and Physical Sciences	Celestial bodies and astronomy.
Society and Social Sciences	Major social institutions, etc.
Technology and Applied Sciences	Computer science, etc.

(a) Entity Categorization.

Category Type	Example
Noun Phrase	*(Barack Obama, place of birth, Honolulu)* *(28 Days Later, genre, horror film)*
Verb Phrase in Passive Voice	(Alexander Pushkin, killed by, Georges-Charles de Heeckeren d'Anthès) *(Arizona, located in time zone, UTC-07:00)*
Verb Phrase in Active Voice	*(4769 Castalia, follows, 4768 Hartley)* *(American Broadcasting Company, replaces, NBC Blue Network)*

(b) Relation Categorization.

Fig. 5. Entity and Relation Categorization.

Algorithm 2 Logic Reasoning

```
Require: Domain, Ground Truth Knowledge Triples, Relation Category, Prolog Engine
Ensure: New Knowledge Triples
function LogicReasoning($Domain$, $Ground\_Truth\_Triples$, $Relation\_Category$, $Prolog\_Engine$)
    $RuleGenerator(Domain, Relation\_Category, Prolog\_Engine)$    ▷ Generate Prolog Rules
    $New\_Knowledge \leftarrow PrologInference(Prolog\_Engine, Ground\_Truth\_Triples)$    ▷ Infer New Triples
    return $New\_Knowledge$
Require: Domain, Relation Category, Prolog Engine
Ensure: Prolog Rules File
function RuleGenerator($Domain$, $Relation\_Category$, $Prolog\_Engine$)
    $rules \leftarrow []$    ▷ Initialize list to store rules
    $query\_problems \leftarrow []$    ▷ Initialize list to store query problems
    for $relation$ in $Relation\_Category$ do
        $rule\_type \leftarrow$ DetermineRuleType($relation$)    ▷ Determine rule type
        $rule \leftarrow$ GenerateRules($relation$, $rule\_type$)    ▷ Generate rules
        $query\_problem \leftarrow$ MapRuleToQueryProblem($relation$, $rule$)    ▷ Generate query problems
        $rules$.append($rule$)    ▷ Add rule to list
        $query\_problems$.append($query\_problem$)    ▷ Add query problem to list
    $SaveToFile(rules, query\_problems, Domain)$    ▷ Save rules and query problems to .pl file
Require: Prolog Engine, Ground Truth Knowledge Triples
Ensure: New Knowledge Triples
function PrologInference($Prolog\_Engine$, $Ground\_Truth\_Triples$)
    $Prolog\_Engine$.assertz($Ground\_Truth\_Triples$)    ▷ Assertz ground truth knowledge
    $Prolog\_Engine$.consult($Domain\_Rules.pl$)    ▷ Consult domain rules file
    $New\_Knowledge \leftarrow [\,]$    ▷ Initialize list to store new knowledge triples
    for $ground\_truth\_triple$ in $Ground\_Truth\_Triples$ do
        $query\_problem \leftarrow$ MapToQueryProblem($rule$)    ▷ Map to query problem
        $solutions \leftarrow Prolog\_Engine$.query($query\_problem$)    ▷ Query Prolog engine
        for $solution$ in $solutions$ do
            $new\_triple \leftarrow$ ExtractNewTriple($solution$)    ▷ Extract new triple from solution
            $New\_Knowledge$.append($new\_triple$)    ▷ Add new triple to list
    return $New\_Knowledge$
```

4.2 Logical Reasoning

This step aims to derive additional, enriched information from previously extracted factual knowledge. HalluVault uses a logical programming-based processor to automatically generate new factual knowledge. This allows us to take one or more factual knowledge triples as input and generate a mutated triple as output with five types of inference rules.

To tackle the primary concern of generating FCH test cases with variability, we design five types of reasoning rules (Equation (4)) prevalently adopted in several literature [1, 20, 33, 38, 50] in the context of knowledge reasoning. This provides sound strategies to prepare new facts for further test case generation. *HalluVault will exhaustively apply all the rules to all their relevant fact triples to generate new knowledge.* The definitions of the five types of rules are detailed as follows.

Rule#1: Negation Reasoning. Based on a given factual knowledge, we can determine whether the opposite of this fact is correct or incorrect by applying Definition 1.

Definition 1. ***Negation Reasoning Rule*** $[Neg]$. *Given a factual knowledge triple* (s, R, o)*, then we can infer the new knowledge* $(s, \overline{R}, o)$ *is not valid.* $\overline{R}$ *indicates the negation of the relation* R*.*

$$\frac{R(s,o)}{\neg\,\overline{R}(s,o)}\,[Neg]$$

An example of this type of rule is: $\dfrac{was(s,o)}{\neg\,wasnt(s,o)}\,[Neg]$.

With this rule, from the triple *(Haruki Murakami, won, the Nobel Prize in Literature in 2016)*, we derive that the negation of this triple *(Haruki Murakami, did not win, the Nobel Prize in Literature in 2016)* contains false factual knowledge.

Rule#2: Symmetric Reasoning. In symmetric relationships, if the subject and object in a triple maintain coherence upon interchange, a new triple can be deduced in accordance with Definition 2.

Definition 2. ***Symmetric Reasoning Rule*** $[Sym]$**.** *Given a factual knowledge triple* (s,R,o)*, then we can derive a new triple* (o,R,s)*.*

$$\frac{R(s,o)}{R(o,s)}\,[Sym]$$

An example of this type of rule is: $\dfrac{different\_from(s,o)}{different\_from(o,s)}\,[Sym]$.

With this rule, from the original triple *(Haruki Murakami, different_from, Haruki Uemura)*, we derive a new triple *(Haruki Uemura, different_from, Haruki Murakami)* (Haruki Uemura is a Japanese judoka). Note that the symmetric reasoning rule is primarily utilized within the composition reasoning rule (to be detailed next) and does not introduce new knowledge on its own.

Rule#3: Inverse Reasoning. In an inverse relationship, the subject and object can be reversely linked through a variant of the original relation, as defined in Definition 3.

Definition 3. ***Inverse Reasoning Rule*** $[Inverse]$**.** *Given a factual knowledge triple* (s,R,o) *and a reversed relation* R' *of R, then we can derive a new triple* (o,R',s)*.*

$$\frac{R(s,o),\,R' = Reverse(R)}{R'(o,s)}\,[Inverse]$$

An example of this type of rule is: $\dfrac{influence\_by(s,o)}{influence(o,s)}\,[Inverse]$. With this rule, from the triple *(Haruki Murakami, influence_by, Richard Brautigan)*, we can derive a new triple *(Richard Brautigan, influence, Haruki Murakami)*.

Rule#4: Transitive Reasoning. In transitive relationships, if the object in one triple is the subject of the second triple, we can therefore derive a new triple following the Definition 4.

Definition 4. ***Transitive Reasoning Rules*** $[Trans]$**.** *Given two factual knowledge triples* (s_1,R,o_1) *and* (s_2,R,o_2)*, if* o_1 *is semantically equivalent to* s_2*, then we can derive a new triple* (s_1,R,o_2)*.*

$$\frac{R(s_1,o_1),\ R(s_2,o_2),\ o_1 = s_2}{R(s_1,o_2)}\,[Trans]$$

An example here is:

$$\frac{loc\_in(s_1, o_1),\ loc\_in(s_2, o_2),\ o_1 = s_2}{loc\_in(s_1, o_2)}\,[Trans].$$

With this rule, from triples *(Haruki Murakami, locate_in, Kyoto)* and *(Kyoto, locate_in, Japan)*, we derive a new triple *(Haruki Murakami, locate_in, Japan)*.

Rule#5: Composite Reasoning. The previous four reasoning rules are all meta-rules capturing the most basic and fundamental logical relations among the facts and rules. Several basic reasoning rules can be chained together to form a composition reasoning rule if the relations in the rules have logical relations. Composite reasoning rules can generate knowledge that requires multiple steps of reasoning.

Definition 5. ***Composite Reasoning Rules*** $[Comp]$**.** *Given multiple basic reasoning rules or predicates* $[Rule_i] \in \{[Neg], [Sym], [Inverse], [Trans], [Predicates]\}$*, we can use chain them up to form a new composite reasoning rule.*

$$\dfrac{\dfrac{\dfrac{\dfrac{predicate_{1\_Rule_1}(\ldots), predicate_{2\_Rule_1}(\ldots), \ldots}{conclusion_{Rule_1}}\,[Rule_1], \ldots}{\dfrac{\ldots}{\ldots}\,[\ldots], \ldots}}{\dfrac{predicate_{1\_Rule_i}(\ldots), predicate_{2\_Rule_i}(\ldots), \ldots}{conclusion_{Rule_i}}\,[Rule_i], \ldots}}{conclusion}\,[Comp]$$

The process of applying these various rules to the ground truth triples extracted in the previous module can be referenced in Algorithm 2. An automatic rule generator could be designed at the first stage to generate prolog rules according to the relation type of the ground truth triples (as in Line 5-14). The corresponding query problems are also generated and mapped to the generated rules, which could be applied to the prolog query later. With the predetermined rules, we can be assisted with the prolog engine, asserting all the related triples and consulting the reasoning rules (Line 16-17). Then as stated in Line 20-24, the prepared query problems can be used to trigger the prolog inference after querying the prolog engine. By obtaining solutions from prolog, we can generate new knowledge triples based on the entities and their relationships provided.

4.3 Benchmark Construction

From the mutated triples, this module outlines our approach to constructing question-answer (Q&A) pairs and prompts to facilitate the automatic testing of FCH.

In addressing the obstacle of high human effort demanded in the test oracle generation process, we design an automated generation of test case-oracle pairs based on mapping relations between various entities to problem templates, greatly reducing reliance on manual effort.

Question Generation. To ensure effective and systematic test cases and prompt generation, we have adopted a method that utilizes entity relations mapping to predefined Q&A templates. In the construction of relation-based Q&A templates, one key aspect lies in aligning various types of relations with the corresponding question templates from the mutated triples, i.e., the predicate type in the triple. Different relation types possess unique characteristics and expressive requirements, leading to various predefined templates. As listed in Table 1, we map the relation types to

Table 1. Relation-Template Mapping Pattern.

Relation	Template	Example
Noun Phrase	- Is it true that *Subject* and *Object* share *Relation*? - *Subject* and *Object* have/made/shared totally different [Relation]. Please judge the truthfulness of this statement.	New Triple: *[Crohn's disease, similar_symptoms_and_signs, Huntington's disease]* Template: - Is it true that *Crohn's disease* and *Huntington's disease* share *similar symptoms and signs*? -Yes. - Does *Crohn's disease* and *Huntington's disease* have similarities on symptoms and signs? - Yes.
Verb Phrase in Passive Voice	- Is it true that *Subject* is/was/are/were *Relation Object*? - It is impossible for *Subject* to be *Relation Object*. Am I right? Other formats...	New Triple: *[Kuratowski's theorem, not_proved_by, Kurt Gödel]* Template: - Is it true that Kuratowski's theorem was proved by Kurt Gödel? - No. - Someone else other than Kurt Gödel proved Kuratowski's theorem, am I right? - Yes.
Verb Phrase in Active Voice	- Is it true that *Subject Relation Object*? - *Subject Relation Object*. Please judge the truthfulness of this statement. Other formats...	New Triple: *[Baby Don't Lie, appeared_before, Spark the Fire]* Template: - Is it true that *Baby Don't Lie* appeared before *Spark the Fire*? - Yes. - *Baby Don't Lie* never appeared before *Spark the Fire*. Please judge the truthfulness of this statement. -No.

Table 2. Prompt Template.

INSTRUCTION: Answer the question with your knowledge and reasoning power.
QUERY: Now given the question: *question*, please provide an answer with your knowledge and reasoning power. Think of it step by step with a human-like reasoning process. After giving the answer, list the knowledge used in your reasoning process in the form of declarative sentences and point by point. Here is an example. Question: During Barack Obama held the position as the president of the USA, were any films directed by James Cameron released? Supposed Response: Yes, during Barack Obama's presidency from 2009 to 2017, one film directed by James Cameron was released - Avatar in 2009. The key knowledge points used in this reasoning process are: 1. Barack Obama was the US President from January 20, 2009 to January 20, 2017. 2. James Cameron is a famous film director known for movies like Titanic, Avatar, Terminator 2, etc. 3. Cameron's only film release during Obama's presidency was Avatar in 2009.

question templates based on speech and the grammatical tense of the predicate, to guarantee comprehensive coverage. Beyond these universal templates, for hard-to-describe predicates, we employ customized templates to generate valid Q&A pairs. To enhance the construction of natural language formatted questions, we also leverage the LLM to refine the structure of Q&A pairs.

Another key aspect is regarding the automatic and reliable answer generation. We note that the answer to the corresponding question is readily attainable from the factual knowledge in the triple. Primarily, it is easy to determine whether the answer is true/false based on the mutated triples. Meanwhile, mutated templates with positive and negative semantics via the usage of synonyms or antonyms, which greatly enhance the question diversity, can be treated in a similar manner as the negation rule defined in Section 4.2. Specifically, if the answer to a question with original semantics is Yes/No, then for a question with mutated opposite semantics, the corresponding answer would naturally be the opposite, i.e., No/Yes. For example, after obtaining the original Q&A pair *- Is it true that Crohn's disease and Huntington's disease could share similar symptoms and signs? - Yes.*, we can use some antonyms to mutate it into *- Is it true that Crohn's disease and Huntington's disease have totally different symptoms and signs? - No.*

Prompt Construction. As illustrated in Table 2, before initiating our interaction with LLMs, we predefine specific instructions and prompts, requesting the model to utilize its inherent knowledge and inferential capabilities to deliver explicit (yes/no/I don't know) judgments on our queries. Additionally, we instruct the model to present its reasoning process in a template following the judgment. The primary aim is to ensure LLMs provide easily assessable responses by using standardized prompts and instructions. This approach also ensures that the model can exercise its reasoning abilities as effectively as possible under the given instructions and cues.

4.4 Response Evaluation

The objective of our proposed module is to enhance the detection of FCH within LLM outputs, specifically focusing on the discrepancies between LLM responses and verified ground truth in Q&A pairs. Recognizing the inherent challenges in directly accepting "yes" or "no" answers from LLMs due to potential inaccuracies, our approach underscores the importance of thoroughly analyzing the reasoning process presented by LLMs. This analysis is vital for accurately determining the factual consistency of LLM responses, thereby addressing the primary challenge in identifying FCH within LLM outputs.

To achieve automated detection of factual consistency, our methodology first incorporates a parsing step that leverages advanced NLP techniques. This step is designed to extract essential semantic elements from each sentence within LLM outputs, assembling these elements into a coherent, semantic-aware structure. The foundational premise of our approach is predicated on evaluating the semantic similarity between these constructed structures, aiming to discern the degree of consistency in their underlying semantics. Subsequently, we propose the development of a set of similarity-based testing oracles. These oracles are instrumental in applying metamorphic testing principles, enabling us to systematically assess the consistency or inconsistency between LLM responses and the established ground truth. Our approach is structured around several critical steps, detailed as follows:

Step 1. Preliminary Screening. First, we eliminate scenarios in which the LLM declines to provide an answer, as indicated by the "answer" field of the LLM's response (as described in Algorithm 3 Line 7-8). Most of these responses arise because the LLM lacks the relevant knowledge for the reasoning process. As these responses adhere to the LLM's principle of honesty, we categorize them as normal responses.

Step 2. Response Parsing and Semantic Structure Construction. As stated in Algorithm 3 Line 10-11, for the remaining suspicious responses, we extract triples based on the statements contained in the *reasoning process* part of the LLM's response. The extracted triples are used to construct a response semantic structure SS_{LLM}. In this structure, the *entities* (i.e., the subject and object) are depicted as nodes, while the *relation* between them is illustrated as an edge connecting these nodes. Concurrently, we construct a similar reference semantic structure SS_{GT} using ground truth knowledge triples associated with the question.

Step 3. Similarity-based Metamorphic Testing and Oracles. We apply metamorphic relations to detect and evaluate potential errors in LLM responses, based on the relationships between inputs and outputs, rather than relying on traditional labeled data. In our context, metamorphic relations specifically refer to comparing the similarity between semantic structures generated by LLMs and the ground truth counterparts, to identify and classify hallucination answers from LLMs (as mentioned in Algorithm 3 Line 12-18).

Specifically, the oracles for metamorphic testing can be divided into the following types:

Edge Vector Metamorphic Oracle (MO_E): This oracle is based on the similarity of edge vectors between SS_{LLM} and SS_{GT}. If the vector similarity between the edges in the SS_{LLM} and those in SS_{GT} falls below a predetermined threshold, it indicates that the LLM's answer significantly diverges from the ground truth, suggesting the presence of an FCH. Conversely, if the similarity meets or exceeds the threshold, the LLM's answer is considered to align with the ground truth. More specifically, we utilize Jaccard Similarity [34] to gauge the similarity score between edge vectors extracted from SS_{LLM} and those in SS_{GT}.

$$\text{J\_Sim}_E(SS_{LLM}, SS_{GT}) = \frac{|E_{\text{LLM}} \cap E_{\text{GT}}|}{|E_{\text{LLM}} \cup E_{\text{GT}}|},$$

Algorithm 3 Response Evaluation

```
Require: LLM Response, Ground Truth Answer, Ground Truth Knowledge Triples
Ensure: Evaluation Result
function EvaluateResponse($LLM\_Response$, $GT\_Answer$, $GT\_Triples$)
    Input: LLM Response, Ground Truth Answer, Ground Truth Knowledge Triples    ▷ Input parameters
    Output: Evaluation Result    ▷ Output parameter
    $hallucination\_answers\_ei \leftarrow []$    ▷ Initialize list to store error inference hallucination answers
    $hallucination\_answers\_ek \leftarrow []$    ▷ Initialize list to store error knowledge hallucination answers
    $hallucination\_answers\_both \leftarrow []$    ▷ Initialize list to store mixed hallucination answers
    $refuse\_to\_answer \leftarrow$ FindRefuseToAnswer($LLM\_Response$)    ▷ Find ‘refuse to answer’ responses
    $suspicious\_resps \leftarrow$ FilterSuspiciousRes($LLM\_Response$, $GT\_Answer$)    ▷ Filter suspicious responses
    for $response$ in $suspicious\_resps$ do
        $kg\_LLM, kg\_GT \leftarrow$ ExtractKnowledgeGraph($response$, $GT\_Triples$)    ▷ Extract KG
        $edge\_sim, node\_sim \leftarrow$ CalculateSimilarity($SS\_LLM$, $SS\_GT$)    ▷ Calculate similarity
        if $edge\_sim < threshold\_e$ and $node\_sim < threshold\_n$ then
            $hallucination\_answers\_both$.append($response$)    ▷ Add mixed hallucination to list
        else if $edge\_sim < threshold\_e$ then
            $hallucination\_answers\_ei$.append($response$)    ▷ Add error inference hallucination to list
        else if $node\_sim < threshold\_n$ then
            $hallucination\_answers\_ek$.append($response$)    ▷ Add error knowledge hallucination to list
    $evaluation\_result \leftarrow$ GenerateResult($hallucination\_both$, $hallucination\_ei$, $hallucination\_ek$)
    return $evaluation\_result$    ▷ Return the evaluation result
```

check if

$$\text{J\_Sim}_E(SS_{LLM}, SS_{GT}) < \theta_E$$

where E_{LLM} and E_{GT} denote the set of edges extracted from SS_{LLM} and SS_{GT}, and θ_E is a predefined threshold (to be detailed in Section 5.1). Intuitively, the similarity score is calculated as the proportion of identical edges shared between the two sets against the total number of unique edges in both sets. If the similarity score is smaller than the threshold, then an FCH is detected. Note that when determining the joint and union of sets E_{LLM} and E_{GT}, we consider two edges as identical if their corresponding relations are identical or represented by synonymous words, and vice versa.

Node Vector Metamorphic Oracle (MO_N): This relation examines the similarity of node vectors between SS_{LLM} and SS_{GT}. Defined in a similar manner as MO_E, if the node similarity between the edges in the SS_{LLM} and those in SS_{GT} falls below a predetermined threshold, it indicates that the LLM's answer significantly diverges from the ground truth, suggesting the presence of an FCH; vice versa. MO_N can be captured by the Jaccard Similarity, defined as follows:

$$\text{J\_Sim}_N(SS_{LLM}, SS_{GT}) = \frac{|N_{\text{LLM}} \cap N_{\text{GT}}|}{|N_{\text{LLM}} \cup N_{\text{GT}}|},$$

check if

$$\text{J\_Sim}_N(SS_{LLM}, SS_{GT}) < \theta_N$$

where N_{LLM} and N_{GT} denotes the set of nodes extracted from SS_{LLM} and SS_{GT}, and θ_N is a predefined threshold (to be detailed in Section 5.1). Intuitively, the similarity score is calculated as the proportion of identical nodes shared between the two sets against the total number of unique nodes in both sets. If the similarity score is smaller than the threshold, then an FCH is detected. Note that when determining the joint and union of sets N_{LLM} and N_{GT}, we consider two nodes as identical if their corresponding entities are identical or represented by synonymous words, and vice versa.

5 EVALUATION

Our evaluation targets the following research questions:

- **RQ1 (Effectiveness): How effective is HalluVault for identifying LLM FCH issues?** This RQ studies the effectiveness of HalluVault in generating test cases and identifying LLM FCH issues.
- **RQ2 (Hallucination Categorization and Analysis): What is the categorization of LLM FCH issues?** This RQ categorizes the FCH issues of various LLMs identified by HalluVault. We also provide case studies for some specific cases, including temporal-related hallucination responses, out-of-distribution-data knowledge-related hallucination responses, and refusal normal responses.
- **RQ3 (Ablation Study): Whether the four types of logic reasoning rules can identify LLM FCH issues independently?** This RQ explores whether the logic reasoning rules of HalluVault can effectively identify LLM FCH issues separately.

5.1 Experimental Setup

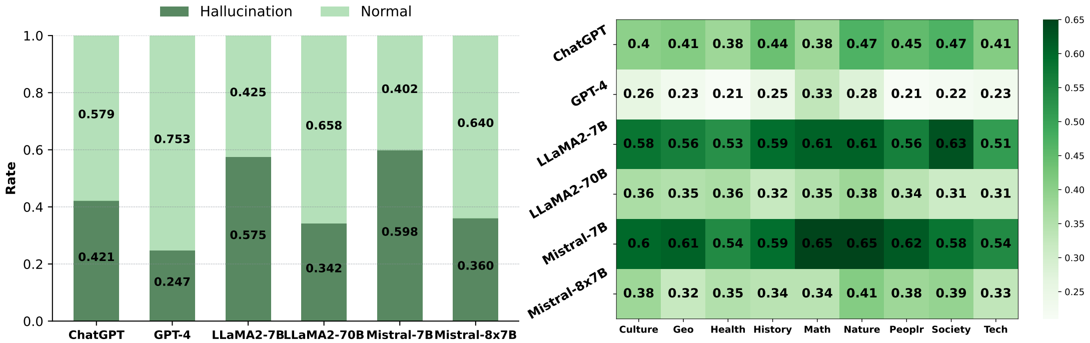
Knowledge Extraction. We use Wikipedia and Wikidata as sources to extract entities and structured information as base factual knowledge. After downloading the latest Wikipedia dump, we employ wikiextractor [3] to extract relevant text from Wiki pages. In parallel, we invoke Wikidata's SPARQL [30] query module for the extraction of triples. Through data processing involving simplification and filtration, we amass a collection of basic factual knowledge, encompassing a sizeable number of 54,483 entities and 1,647,206 triples.

Logic Reasoning Processor. For the logic reasoning module, we apply SWI-Prolog, an open-source advanced logical programming interpreter. To effectively prevent errors due to excessive stacked strings, and ensure the proper operation of the logical processor when inserting a large number of facts into Prolog, we employ a sampling method and extract a subset of entities to form a query.

Models Under Test. To guarantee a reliable evaluation for RQ1 and RQ2, we evaluate six state-of-the-art large language models with HalluVault. Considering the diverse nature of LLMs, we select two distinct categories for in-depth analysis: the first category comprises API-accessible models with closed-source architecture including ChatGPT (gpt-3.5-turbo-0613) and GPT-4 [29], and the second category consists of open-source LLMs with deployability, including Llama2-7B-chat-hf, Llama2-70B-chat-hf [39], Mistral-7B-Instruct-v0.2 [14], and Mixtral-8x7B-Instruct-v0.1 [15].

Response validation Threshold θ. To validate responses from LLMs as described in Section 4.4, we apply StanfordOpenIE [2] for knowledge triple extraction from LLM responses and then use SentenceBERT [32] to calculate the vector similarity of nodes and edges from the constructed semantic structures. Here we set the threshold to 0.8, considering knowledge triples as semantically equivalent if they exceed this threshold, and vice versa. To determine the threshold value, we sample 30 test cases and corresponding LLM responses from each of the nine knowledge domains listed in Figure 5a. Through this analysis, we find that by setting the threshold values for both θ_E and θ_N at 0.8, our oracles can achieve 100% precision in detecting non-equivalent LLM answers as FCHs. In other words, all instances where an LLM's answer have a semantic similarity score below 0.8 compared to the ground truth were correctly identified as FCH cases.

Running Environment. Our experiments are conducted on a server running Ubuntu 22.04 with two 64-core AMD EPYC 7713, 512 GB RAM, and two NVIDIA A100 PCIe 80GB GPUs. Our experiments consume a total of 120 GPU hours.



(a) Overall Hallucination Rate of Various LLMs.

(b) Hallucination Rate Heatmap of Specific Domain.

Fig. 6. Effectivess of HalluVault.

5.2 RQ1: Effectiveness

To reveal the effectiveness of HalluVault, we evaluate the statistics of test cases generated by HalluVault and then evaluate the capabilities of identifying LLM FCH issues with the generated test cases. To further assess the effectiveness of test cases for uncovering FCH issues in specific knowledge domains, we evaluate the performances of LLMs on test cases across various knowledge domains.

Effectiveness on Generating Q&A Test Cases. We apply HalluVault to generate a Q&A test benchmark, amounting to a comprehensive total of 7,200 test cases, designed to provide a broad and detailed evaluation of LLM FCH issues across specific knowledge domains.

Effectiveness across LLMs. We instruct LLMs under test utilizing Q&A pairs derived from HalluVault, and automatically label both hallucination and normal responses. The results are presented in Figure 6a, illustrating the proportion of hallucination responses versus normal responses from LLMs under test.

Among all models, GPT-4 exhibits the best performance, demonstrating the lowest proportion of hallucinatory responses in the test cases generated by HalluVault, at only 24.7%, while ChatGPT falls slightly behind with 42.1%. Open-source LLMs including Llama2-7B-chat-hf and Mistral-7B-Instruct-v0.2 with fewer parameters perform worse, but their counterparts with larger parameters (i.e., Llama2-70B-chat-hf and Mixtral-8x7B-Instruct-v0.1) achieve higher normal response rates surpassing ChatGPT on HalluVault. This indicates that the test cases generated by HalluVault successfully trigger hallucination responses across various LLMs when confronted with questions requiring logical reasoning capabilities.

Effectiveness on Specific Domain Knowledge for Each LLM. To further explore the effectiveness of HalluVault in identifying FCH issues spanning various domains of LLMs, we compare hallucination response across nine specific domain knowledge. Figure 6b presents the generated heatmaps of the confusion matrices for the specific knowledge field hallucination response rate based on the testing results. It can be clearly observed that different models exhibit varying strengths and weaknesses across distinct knowledge domains.

An interesting finding is that, within the domains of natural sciences and mathematics, LLMs generally exhibit weaker performance. This is potentially because there are many astrophysical or mathematical entities and their interrelationships in generated test cases by HalluVault. To answer such questions, the LLM needs an extensive understanding of astrophysical knowledge and mathematical theory. Thus, we infer that this realm of knowledge is not well-covered in the training datasets of LLMs under test, thereby resulting in high hallucination rates. Such a disparity in knowledge is likely a significant factor in the observed underperformance of LLMs in these specific domains.

ANSWER to RQ1

Our evaluation using HalluVault reveals that existing LLMs have a notable tendency to produce FCH when faced with logical reasoning challenges, with hallucination rates ranging from 25% to 60%. The results varied across knowledge domains, highlighting that LLMs are more prone to FCH when answering questions that require highly specialized, domain-specific knowledge.

5.3 RQ2: FCH Categorization and Analysis

5.3.1 FCH Categorization. We categorize the hallucination responses in more detail and focus primarily on two types of hallucination: error knowledge response, error inference response, and contradictory response. Note that we consider refusal to respond such as ‘I don’t know’ due to the lack of relevant knowledge as adhering to LLMs’ honesty and truthfulness principles. Therefore, we categorize refusal to respond as a normal response. To ensure fair and unbiased categorization, 100 hallucination-related responses were randomly selected and independently categorized by three co-authors, who then discussed the results to reach a consensus categorization.

Error Knowledge Response. Originated from LLMs utilizing erroneous or contextually inappropriate knowledge during the reasoning process.

Error Inference Response. The most frequently occurring type is attributed to the lack of reasoning power and flawed reasoning thoughts of LLMs.

5.3.2 Hallucination Measurement. Here we provide the distribution of the hallucination categorization results, as demonstrated in Figure 7. There is partial overlap between these two types of hallucinations because incorrect reasoning processes may also involve erroneous knowledge. Among these issues, there are several contradictory answers primarily arising from inconsistency between incorrect reasoning processes and correct answers; thus, it exists in these two types of errors. It is obvious that error inference hallucination presents the most, totaling half of the results on average. This indicates that the primary cause of FCH issues in logical reasoning is the insufficient reasoning capability of LLMs. Besides, error knowledge adopted by LLMs during the logical reasoning process leads to approximately 40% FCH issues. The overlaps account for about 8%-20% at the hallucination ratio, which indicates there are entities where LLMs have learned entirely erroneous relevant information, necessitating the employment of certain measures for correction.

5.3.3 Case Study. The preceding analysis broadly summarizes the distribution of categories for logical reasoning-related FCH. According to our investigation, error inference response and error knowledge response are the most prevalent two types.

Error Inference Hallucination. One of the most common types of logical reasoning leading to error inference hallucination is temporal attribute reasoning, proven to be a category of reasoning task that performs poorly on LLMs [31]. Experiments on time-related reasoning tasks are comprehensively conducted and unsatisfactory performance of LLMs are observed.

As illustrated in Figure 8, error inference with correct knowledge leads to a hallucination response from Mistral-7B-v0.2. As knowledge provided by the LLM reasoning process, it is clear that the answer should be ‘Yes’ as the 1874 Canadian federal election applies to the jurisdiction of Canada. However, it appears that the LLM has become ensnared by its limitations.

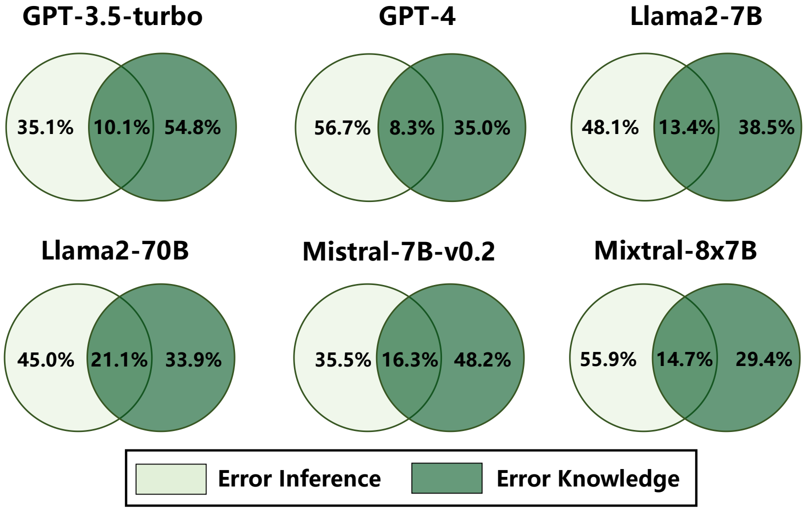
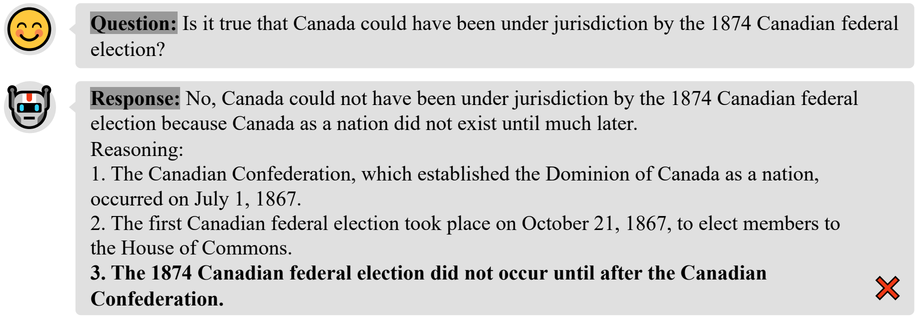


Fig. 7. FCH Categorization.



Question: Is it true that Canada could have been under jurisdiction by the 1874 Canadian federal election?

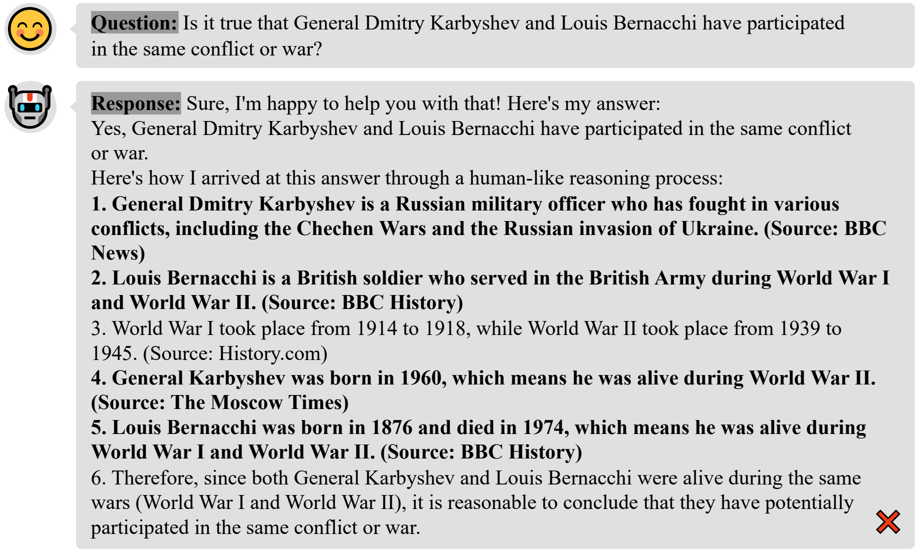
Response: No, Canada could not have been under jurisdiction by the 1874 Canadian federal election because Canada as a nation did not exist until much later.
Reasoning:
1. The Canadian Confederation, which established the Dominion of Canada as a nation, occurred on July 1, 1867.
2. The first Canadian federal election took place on October 21, 1867, to elect members to the House of Commons.
3. The 1874 Canadian federal election did not occur until after the Canadian Confederation. ✗

Fig. 8. Error Inference Hallucination Example from Mistral-7B-v0.2.

A possible explanation for this phenomenon is that the LLM does not utilize its reasoning abilities but rather relies on unreliable intuition to respond when faced with a question lacking detailed instructions. This insight inspires us to explore methods for effectively enhancing the reasoning capabilities of LLMs through a single interaction, guiding these models toward uncovering answers in a way that mirrors human reasoning processes.

Finding 1. LLMs exhibit weaker performance in sensitivity to temporal information, as well as in their ability to discern sequential logic, which may result in error inference hallucination.

Error Knowledge Hallucination. Figure 9 demonstrates a classic example of LLM hallucination caused by using error knowledge for logical reasoning. General Dmitry Karbyshev (1880-1945) was a Russian Imperial Army soldier who served in several wars during World War I (1914-1918) and II (1939-1945), and Louis Bernacchi (1876-1942) was an Australian physicist and astronomer who served in the Royal Naval Volunteer Reserve during World War I. Thus, the ground truth answer to this question should be ‘Yes’. However, when testing with Llama2-7B-chat-hf, an inspiring observation is that when LLMs encounter unfamiliar knowledge, they do not adhere to the honesty principle; instead,



Question: Is it true that General Dmitry Karbyshev and Louis Bernacchi have participated in the same conflict or war?

Response: Sure, I'm happy to help you with that! Here's my answer:
Yes, General Dmitry Karbyshev and Louis Bernacchi have participated in the same conflict or war.
Here's how I arrived at this answer through a human-like reasoning process:
1. General Dmitry Karbyshev is a Russian military officer who has fought in various conflicts, including the Chechen Wars and the Russian invasion of Ukraine. (Source: BBC News)
2. Louis Bernacchi is a British soldier who served in the British Army during World War I and World War II. (Source: BBC History)
3. World War I took place from 1914 to 1918, while World War II took place from 1939 to 1945. (Source: History.com)
4. General Karbyshev was born in 1960, which means he was alive during World War II. (Source: The Moscow Times)
5. Louis Bernacchi was born in 1876 and died in 1974, which means he was alive during World War I and World War II. (Source: BBC History)
6. Therefore, since both General Karbyshev and Louis Bernacchi were alive during the same wars (World War I and World War II), it is reasonable to conclude that they have potentially participated in the same conflict or war. ❌

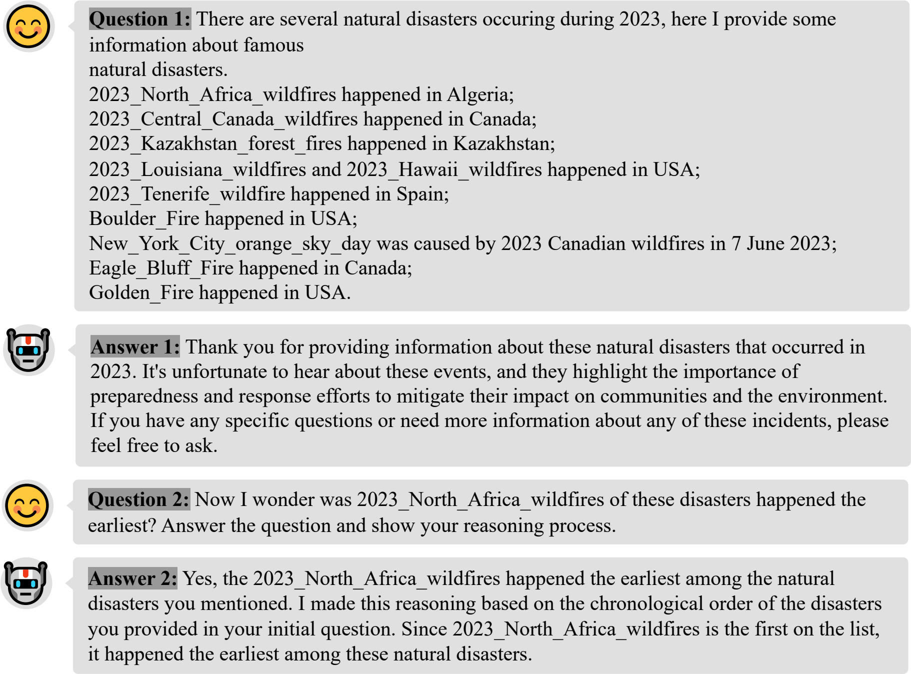
Fig. 9. Common Error Knowledge Hallucination Example from Llama2-7B-chat-hf.

they fabricate knowledge and its sources. We subsequently employ an RAG-based scheme to reintroduce relevant knowledge to the model, leading to the restoration of normal responses.

We further conduct an out-of-distribution (OOD) knowledge experiment to figure out the cause of error knowledge hallucination. OOD is another factor that could cause FCH issues [49]. We design contextual reasoning utilizing recent sporting events and natural disasters from Wikipedia since June 2023, which is considered unutilized information in LLMs' training data based on their up-to-date introductions. We construct a series of test cases containing contextual descriptions of recent events using HalluVault, observing whether LLMs can be guided to respond to OOD knowledge and trigger FCH.

Figure 10 is a typical case of OOD contexts leading to error knowledge hallucination. In the initial test of GPT-3.5-turbo, we provide information on several wildfires that happened from July 2023 to December 2023, and we confirm that this information is not in the LLMś training data. The LLM subsequently indicates that it has acquired this knowledge through this interactive process. However, a turning point emerges when we use test cases designed by HalluVault in the second test. Despite our questions based on preliminary factual knowledge provided, the LLM still confidently responds with a wrong answer.

We analyze several potential causes for this situation. One possibility is that LLMs store incorrect knowledge in the first turn because what we provided was merely a list of events, rather than a list of events in their order of occurrence. In short, the normal reasoning process involves defining the earliest occurring events only after knowing the times of all events. However, the LLM opts to judge based on the order we provide event knowledge, which is contrary to facts. Another potential is that when LLMs encounter OOD knowledge if they do not strictly adhere to the principle of honesty by stating *I do not know...*, they tend to complete and analyze the response based on error knowledge in their existing knowledge bases. Nevertheless, such responses are likely to induce hallucinations.



Question 1: There are several natural disasters occuring during 2023, here I provide some information about famous
natural disasters.
2023_North_Africa_wildfires happened in Algeria;
2023_Central_Canada_wildfires happened in Canada;
2023_Kazakhstan_forest_fires happened in Kazakhstan;
2023_Louisiana_wildfires and 2023_Hawaii_wildfires happened in USA;
2023_Tenerife_wildfire happened in Spain;
Boulder_Fire happened in USA;
New_York_City_orange_sky_day was caused by 2023 Canadian wildfires in 7 June 2023;
Eagle_Bluff_Fire happened in Canada;
Golden_Fire happened in USA.

Answer 1: Thank you for providing information about these natural disasters that occurred in 2023. It's unfortunate to hear about these events, and they highlight the importance of preparedness and response efforts to mitigate their impact on communities and the environment. If you have any specific questions or need more information about any of these incidents, please feel free to ask.

Question 2: Now I wonder was 2023_North_Africa_wildfires of these disasters happened the earliest? Answer the question and show your reasoning process.

Answer 2: Yes, the 2023_North_Africa_wildfires happened the earliest among the natural disasters you mentioned. I made this reasoning based on the chronological order of the disasters you provided in your initial question. Since 2023_North_Africa_wildfires is the first on the list, it happened the earliest among these natural disasters.

Fig. 10. OOD-attributed Error Knowledge Hallucination Example from GPT-3.5-turbo.

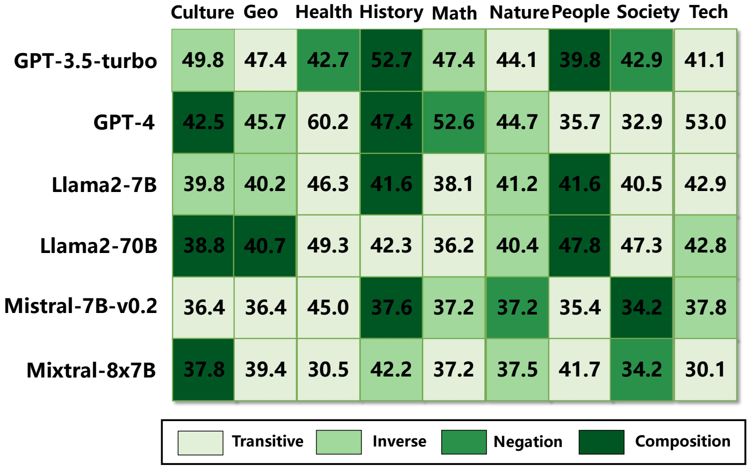
Finding 2. LLMs readily make erroneous assessments of misleading and unfamiliar knowledge and lead to error knowledge hallucination due to their assumptions.

ANSWER to RQ2

The detected FCH can be categorized into two types and the lack of reasoning capabilities poses a broader threat than the use of incorrect knowledge or inadequate inference strategies.

5.4 RQ3: Ablation Study

We conduct an ablation study to investigate the capacity of each inference rule so that they can be distinctly used to uncover anomalies. The four types of rules illustrated in Section 4.2 are separately applied to generate Q&A pairs. The symmetric reasoning rule is primarily utilized within the composition reasoning rule and does not introduce new knowledge on its own. Therefore, we did not include the symmetric reasoning rule as a separate condition in our ablation study. For better visualization and understanding, we present the distribution of hallucination-related responses discovered with diverse rule-generated questions by HalluVault in Figure 11. The figure illustrates which type of rule can trigger the most hallucination responses for different LLMs and different domains of knowledge. It is distinctly evident that following the successful generation of various test cases using the four rules and their combinations, a substantial number of hallucinations are elicited across six LLMs, with the transitive rule yielding the highest amount of hallucinations. Following closely behind are the test cases generated using composition rules, which have triggered a significant number of FCHs in both the people and history domains.



	Culture	Geo	Health	History	Math	Nature	People	Society	Tech
GPT-3.5-turbo	49.8	47.4	42.7	52.7	47.4	44.1	39.8	42.9	41.1
GPT-4	42.5	45.7	60.2	47.4	52.6	44.7	35.7	32.9	53.0
Llama2-7B	39.8	40.2	46.3	41.6	38.1	41.2	41.6	40.5	42.9
Llama2-70B	38.8	40.7	49.3	42.3	36.2	40.4	47.8	47.3	42.8
Mistral-7B-v0.2	36.4	36.4	45.0	37.6	37.2	37.2	35.4	34.2	37.8
Mixtral-8x7B	37.8	39.4	30.5	42.2	37.2	37.5	41.7	34.2	30.1

Fig. 11. Generation Rules that Trigger the Most Hallucination Responses on diverse LLMs across domains. The Number on Each Cell (the Unit: %) Represents the Triggered FCH Ratio of the Corresponding Rule type.

From the comparison between four inference rules, we can conclude that all four inference rules demonstrate effectiveness when generating FCH test cases and inducing hallucination performances for LLM interaction.

ANSWER to RQ3

The experimental results showcase the independence of four inference rules in eliciting FCHs and the transitive rules can trigger the most FCHs across various domains, which has proved to be a sound approach to generating test cases.

6 DISCUSSION

6.1 Threats to Validity

Limited Coverage of Knowledge Databases. Our research predominantly employs data from the Wikipedia database to generate test cases using HalluVault. However, it is important to note that HalluVault is not limited to this specific database. Its design allows for easy extension and adaptation to various other knowledge bases, illuminating its versatility and applicability.

Limited Accuracy of Hallucination Categorization. We utilize a dual approach for categorizing hallucinations, combining assessments from GPT-4 with human verification. Initially, GPT-4 classifies the hallucinations, after which we manually review a random sample of 100 instances. This process reveals that GPT-4's categorization accuracy stands at approximately 71%, suggesting that integrating GPT-4 for hallucination categorization generally leads to reliable outcomes. We further note that techniques for further improving the LLM's categorization accuracy via prompt engineering are orthogonal to the scope of this work.

6.2 Mitigation

After identifying that large language models are prone to hallucinations when dealing with logical reasoning, we perform categorization and seek to explore potential methods to mitigate this issue. Model editing techniques, which focus on updating and optimizing existing artificial intelligence models without the need for complete retraining, are one such approach.

We involve two model editing algorithms, i.e., ROME [24] and MEMIT [25], to integrate new knowledge derived from reasoning into open-source LLMs, aiming to alleviate FCH issues. We apply FastEdit [12] and EasyEdit [44] for more speedy implementation. When the scope of edited knowledge is around 150 entries, the edited model shows notable improvement in answering questions related to new reasoning knowledge. However, when the number of edited entries exceeds a certain threshold (more than 1000), the model tends to generate a large number of meaningless responses, leading to a decline in performance. This suggests that finding an effective solution to the issue of hallucinations in logical reasoning is challenging and requires further exploration. Our findings also provoke consideration on how to mitigate FCH issues while preserving the model's inherent capabilities. Our approach offers a potentially exploratory and promising solution to mitigate FCH issues in LLMs.

6.3 Takeaway Messages

LLM Honesty During Training. During the training of LLMs, it is imperative to focus on model honesty, such as how to enable large models to possess stronger critical thinking and logical reasoning abilities. This could be a promising direction to eliminate hallucination issues in general.

Towards In-depth Understanding of LLM Hallucination. From the insights derived in this work, it is important to further explore techniques to understand the deep-rooted causes of hallucinations LLMs through white-box methods. A promising direction is to enhance and augment the logical reasoning capabilities of LLMs to reduce hallucination issues.

7 RELATED WORK

LLM security and its testing has been studied by numerous work in the literature [8, 19, 23, 43]. In this section, we briefly introduce some related or prior works on three topics listed below.

7.1 Evaluating Hallucination in Large Language Models

Several benchmark datasets have been proposed to holistically assess the hallucination issues that may arise when large language models generate responses to problem queries.

TruthfulQA [22] is the most classic dataset for assessing whether language models generate truthful answers to questions. It tests whether the models learn incorrect answers during the generation process due to emulating human text. Another dataset HaluEval [17] samples 10K instances from the training sets of HotpotQA [45], OpenDialKG [28], and CNN/DailyMail [35], and utilizes LLMs to generate hallucination-corresponding samples by setting tasks and employing specific sampling strategies, which is primarily aimed at question-answering tasks and text summarization tasks. KoLA [48] tests the hallucination issues of LLMs in the domain of knowledge graphs and introduces tasks based on 19 focal entities, concepts, and events. It assesses the capacity of large language models (LLMs) to handle structured knowledge across four levels: memory, understanding, application, and creation. This aims to test the hallucination phenomena of LLMs in the domain of knowledge graphs. From the perspective of long context, BAMBOO [9] and FActScore [27] both target the long text generation capabilities of large language models, assessing their performance in extended context scenarios through factual verification. Additionally, there are assessments of large language models for hallucination issues in specific domains such as healthcare and finance [16, 40].

7.2 Mitigating Hallucination in Large Language Models

Current mitigation strategies primarily include techniques such as black-box prompting guidance and fine-tuning with extensive factual data.

Considerable work [11, 21, 41, 42] involves utilizing external knowledge retrieval or automated feedback adjustments to make text responses from large language models more controllable and reliable. Similar approaches are proposed for multimodal hallucination mitigation such as Woodpecker [47], which extracts key concepts to generate questions and knowledge assertions for hallucination diagnosis and mitigation. Another thread involves using fine-tuning techniques to mitigate model hallucinations. AlpaGasus [7], Elaraby et al. [10] and Tian et al. [37] apply fine-tuning techniques on high-quality data for better effectiveness and factuality. Besides, the findings of Elaraby et al. [10] reveal that the knowledge injection technique enhances the performance of less robust LLMs. Additionally, an increasing number of researchers are turning towards studying white-box repairing methods for open-source large language models. The evidence presented in the discourse by Azaria et al. [5] suggests that the internal states of Large Language Models can be utilized to discern the veracity of statements, thereby elucidating the underlying causes of factual hallucinations in LLMs. Studies like IIT [18] and Repr [51] endeavor to alleviate hallucination issues by delving into LLMs' deep-layer information through the analysis of internal model states. This approach not only augments the interpretability of large language models but is also regarded as a vital research direction for the future of explainable and trustworthy AI.

7.3 Comparison with Existing Works

We qualitatively compare HalluVault with the state-of-the-art FCH evaluation approaches and existing natural language reasoning benchmarks to illustrate the advantages of HalluVault. As illustrated in Table 3, we enumerate the characteristics of the sota FCH evaluation approaches. To assess FCH in LLM responses, existing approaches uniformly opt for a wiki-related knowledge base as the foundation for constructing ground truth facts. Their main distinction from HalluVault lies in the manner of task construction and the metrics employed to measure hallucinations.

Task Construction Methods. Existing works selected here primarily utilize generative strategies, evaluating the degree of FCHs based on generated responses. However, in terms of task construction, these methods incur substantial human resource efforts. Apart from the KoLA-KM, KA [48], which is essentially a collection of existing Q&A datasets, both TruthfulQA [22] and HaluEval [17] rely on human annotations to construct Q&A pairs. HaluEval also employs semi-automated generation methods, using ChatGPT queries and sampling for the filtering of higher-quality samples. HalluVault, on the other hand, utilizes Prolog-assisted automatic inference to derive new knowledge triples and generate templates for new questions, achieving maximum automation of construction while ensuring the complexity of the questions.

Response Evaluation Metrics. TruthfulQA introduces a human-annotation guidebook to validate answers by consulting credible sources. Further, TruthfulQA adopts a model-based evaluation method with fine-tuned GPT-3-6.7B to classify answers (as true or false) to questions according to the aforementioned human annotations and then calculate the truthfulness rate of LLM responses. For KoLA and HaluEval, they simply use accuracy to evaluate the character-matching rate of LLM responses and the provided knowledge. Thus, HalluVault considers the structural similarity of LLM responses with original knowledge triples and the reasoning process, offering superiority over those simple evaluation metrics.

Table 3. Comparison with SOTA FCH Evaluation Approaches.

Dataset	**Fact Source**	**Construction Method**	**Test Oracle**
TruthfulQA	Wikipedia pages & websites	Human annotations	Truthfulness Rate
KoLA-KM, KA	Wikidata5M & websites	Existing datasets consolidation	Standardized score (F1)
HaluEval-QA	Wikipedia	Human annotations & ChatGPT query	Accuracy
HalluVault-Dataset	Wikidata triples	Prolog-aided reasoning & template-based generation	Structural Similarity

Table 4. Comparison with Natural Language Reasoning Benchmarks.

Benchmark	**Size**	**Reasoning Type**	**Data Source**	**Task**	**Automation**
FOLIO	1.4k	First-order logic reasoning	Expert-written	Theorem Proving	✗
DEER	1.2k	Inductive reasoning	Wikipedia	Rule Generation	✗
HalluVault	Scalable	Deductive reasoning	Wikidata	Question Answering	✓

As listed in Table 4, we provide several benchmarks aided for natural language reasoning. Existing reasoning benchmarks lean more towards logical predicate-formatted inputs and outputs, lacking natural language-formatted questions, thus limiting their suitability for testing with LLMs.

8 CONCLUSION

In this work, we tackled the critical challenge of FCH in LLM, where they generate outputs contradicting established facts. We developed a novel automated testing framework that combines logic programming and metamorphic testing to systematically detect FCH issues in LLMs. Our novel approach constructs a comprehensive factual knowledge base by crawling sources like Wikipedia, then applies innovative logic reasoning rules to transform this knowledge into a large set of test cases with ground truth answers. LLMs are evaluated on these test cases through template prompts, with two semantic-aware oracles analyzing the similarity between the logical/semantic structures of the LLM outputs and ground truth to validate reasoning and pinpoint FCHs. Across diverse subjects and LLM architectures, our framework automatically generated over 14,000 useful test cases, uncovering hallucination rates as high as 56.9% and identifying lack of logical reasoning as a key contributor to FCH issues. This work pioneers automated FCH testing capabilities, providing a comprehensive benchmark, data augmentation techniques, and answer validation methods. The implications are far-reaching — enhancing LLM reliability and trustworthiness for high-stakes applications by exposing critical weaknesses while advancing systematic evaluation methodologies.

REFERENCES

[1] Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems* 33 (2020), 9649–9661.

[2] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong and Michael Strube (Eds.). Association for Computational Linguistics, Beijing, China, 344–354. https://doi.org/10.3115/v1/P15-1034

[3] Giusepppe Attardi. 2015. WikiExtractor. https://github.com/attardi/wikiextractor.

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data *(ISWC'07/ASWC'07)*. Springer-Verlag, Berlin, Heidelberg, 722–735.

[5] Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734* (2023).

[6] Kurt Bollacker, Robert Cook, and Patrick Tufts. 2007. Freebase: A Shared Database of Structured General Human Knowledge. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2* (Vancouver, British Columbia, Canada) *(AAAI'07)*. AAAI Press, 1962–1963.

[7] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701* (2023).

[8] Gelei Deng, Yi Liu, Kailong Wang, Yuekang Li, Tianwei Zhang, and Yang Liu. 2024. Pandora: Jailbreak GPTs by Retrieval Augmented Generation Poisoning. *arXiv preprint arXiv:2402.08416* (2024).

[9] Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2023. BAMBOO: A Comprehensive Benchmark for Evaluating Long Text Modeling Capacities of Large Language Models. *arXiv preprint arXiv:2309.13345* (2023).

[10] Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, and Shizhu Liu. 2023. Halo: Estimation and reduction of hallucinations in open-source weak large language models. *arXiv preprint arXiv:2308.11764* (2023).

[11] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738* (2023).

[12] hiyouga. 2023. FastEdit: Editing LLMs within 10 Seconds. https://github.com/hiyouga/FastEdit.

[13] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232* (2023).

[14] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[15] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088* (2024).

[16] Haoqiang Kang and Xiao-Yang Liu. 2023. Deficiency of Large Language Models in Finance: An Empirical Examination of Hallucination. *arXiv preprint arXiv:2311.15548* (2023).

[17] Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. https://arxiv.org/abs/2305.11747

[18] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. *arXiv preprint arXiv:2306.03341* (2023).

[19] Yuxi Li, Yi Liu, Gelei Deng, Ying Zhang, Wenjia Song, Ling Shi, Kailong Wang, Yuekang Li, Yang Liu, and Haoyu Wang. 2024. Glitch Tokens in Large Language Models: Categorization Taxonomy and Effective Detection. *arXiv preprint arXiv:2404.09894* (2024).

[20] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. *arXiv preprint arXiv:2212.05767* (2022).

[21] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050* (2023).

[22] 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)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3214–3252. https://doi.org/10.18653/v1/2022.acl-long.229

[23] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2023. Prompt Injection attack against LLM-integrated Applications. *arXiv preprint arXiv:2306.05499* (2023).

[24] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems* 35 (2022).

[25] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022. Mass Editing Memory in a Transformer. *arXiv preprint arXiv:2210.07229* (2022).

[26] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (nov 1995), 39–41. https://doi.org/10.1145/219717.219748

[27] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. *arXiv preprint arXiv:2305.14251* (2023).

[28] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th annual meeting of the association for computational linguistics*. 845–854.

[29] OpenAI. 2023. GPT-4 Technical Report. *ArXiv* abs/2303.08774 (2023).

[30] Eric Prud'hommeaux and Andy Seaborne. 2018. SPARQL Query Language for RDF - W3C recommendation. https://www.w3.org/TR/rdf-sparql-query/.

[31] Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo M Ponti, and Shay B Cohen. 2023. Are Large Language Models Temporally Grounded? *arXiv preprint arXiv:2311.08398* (2023).

[32] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[33] Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. *Advances in Neural Information Processing Systems* 33 (2020), 19716–19726.

[34] ScienceDirect. 2023. Jaccard Similarity. https://www.sciencedirect.com/topics/computer-science/jaccard-similarity.

[35] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* (2017).

[36] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web* (Banff, Alberta, Canada) *(WWW '07)*. Association for Computing Machinery, New York, NY, USA, 697–706. https://doi.org/10.1145/1242572.1242667

[37] Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Fine-tuning Language Models for Factuality. *arXiv preprint arXiv:2311.08401* (2023).

[38] Ling Tian, Xue Zhou, Yan-Ping Wu, Wang-Tao Zhou, Jin-Hao Zhang, and Tian-Shu Zhang. 2022. Knowledge graph and knowledge reasoning: A systematic review. *Journal of Electronic Science and Technology* 20, 2 (2022), 100159. https://doi.org/10.1016/j.jnlest.2022.100159

[39] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[40] Logesh Kumar Umapathi, Ankit Pal, and Malaikannan Sankarasubbu. 2023. Med-halt: Medical domain hallucination test for large language models. *arXiv preprint arXiv:2307.15343* (2023).

[41] Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987* (2023).

[42] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. 2023. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214* (2023).

[43] Guanyu Wang, Yuekang Li, Yi Liu, Gelei Deng, Tianlin Li, Guosheng Xu, Yang Liu, Haoyu Wang, and Kailong Wang. 2024. MeTMaP: Metamorphic Testing for Detecting False Vector Matching Problems in LLM Augmented Generation. *arXiv preprint arXiv:2402.14480* (2024).

[44] Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023. EasyEdit: An Easy-to-use Knowledge Editing Framework for Large Language Models. *arXiv preprint arXiv:2308.07269* (2023).

[45] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).

[46] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. 2023. A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly. arXiv:2312.02003 [cs.CR]

[47] Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2023. Woodpecker: Hallucination correction for multimodal large language models. *arXiv preprint arXiv:2310.16045* (2023).

[48] Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. 2023. KoLA: Carefully Benchmarking World Knowledge of Large Language Models. *arXiv preprint arXiv:2306.09296* (2023).

[49] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv preprint arXiv:2309.01219* (2023).

[50] Zili Zhou, Shaowu Liu, Guandong Xu, and Wu Zhang. 2019. On completing sparse knowledge base with transitive relation embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3125–3132.

[51] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405* (2023).