POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Measuring Contextual Adherence in LLM-based Reasoning over Knowledge Graphs

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Riccardo Dell'Oro, 10676383

Advisor:
Prof. Marco Brambilla

Co-advisors:
Dr. Antonio De Santis

Academic year:
2025-2026

Abstract: Large Language Models (LLMs) combine parametric knowledge acquired during training with contextual knowledge provided within the prompt, yet they often struggle to override incorrect priors when presented with contradictory evidence. This thesis investigates this "tug-of-war" in the context of Knowledge Graph (KG)-based Retrieval-Augmented Generation (RAG). Using a question-answering dataset accompanied by a Knowledge Graph, four progressively altered variants (Slight, Significant, Comical and Uncomprehensible) are constructed introducing controlled contradictions inside the KG. Three KG-based RAG methods, Graph Constrained Reasoning (GCR), Reasoning on Graphs (RoG) and Think on Graph (ToG), are evaluated in terms of reasoning-path quality, answer accuracy, and adherence to contextual knowledge, measured through Prior Bias and Context Bias. Experimental results show that ToG is unreliable due to frequent path-generation failures, while GCR consistently outperforms RoG by producing only valid paths and achieving higher adherence to contextual information. Although KG-based RAG significantly improves factual grounding compared to a no-retrieval baseline, adherence to contradictory context remains partial and degrades as perturbations increase. These findings highlight both the benefits and the limitations of structured retrieval in guiding LLM reasoning under conflicting knowledge. All the results and the code used are available on GitHub[a].

[a] https://github.com/riccardodelloro/KG-Reasoning-Analysis

Key-words: Large Language Models, Retrieval-Augmented Generation, Knowledge Graphs, Hallucination Mitigation, Contextual Knowledge, Parametric Knowledge.

1. Introduction

When a knowledge-based query is submitted to a Large Language Model (LLM), the output of the model is influenced by two distinct types of knowledge. The first is parametric knowledge [16]; this knowledge is learned during the training phase and is implicitly encoded in the model's weights, rendering it immutable without retraining or fine-tuning the model. The second is contextual knowledge [7]; this knowledge is derived from textual information provided as input to the LLM; this form of knowledge is explicit and can be dynamically modified.
When relying only on their parametric knowledge, LLMs have been found to suffer from hallucinations and

inaccuracies in their responses [1, 9, 15]. The knowledge base provided by these models is limited to their learning dataset; hence, they cannot answer any questions regarding recent facts or any private information.
Retrieval-Augmented Generation (RAG) has proved to be an efficient approach to overcoming these limitations. By incorporating information obtained from an external knowledge source directly into the prompt used in the model, enhancing contextual knowledge, RAG improves factual grounding and overall correctness in responses [4].
However, these scenarios also raise new concerns for LLMs. Suppose that the data obtained from the search query contain incorrect information; in these situations, it is possible for the LLM to report incorrect answers or misleading information obtained from the retrieved documents [5, 14]; there could also be situations where the documents obtained from the search query are correct, but the LLM continues with its incorrect prior knowledge.
The tug-of-war between parametric and contextual knowledge has been studied by different research. Longpre et al.[11] were able to show that LLMs have a tendency to trust their parametric knowledge rather than the contradictory given context. Another research by Xie et al.[22] shows conflicting results, and that the propensity to be influenced by contextual information is affected by the form in which the information is provided to the model. Another research on this topic is the work by Wu et al.[21] with the ClashEval dataset, where they compared the performance of different LLMs when presented with contextual knowledge in contrast to reality, using an artificially altered question-answer dataset.
Knowledge Graphs (KGs) are a data structure used to store conceptual information in a structured format using nodes for entities and edges for relationships. They integrate heterogeneous data, adding semantic meaning and context. Unlike unstructured texts, KGs represent relationships in an explicit manner. There have been efforts in adopting KGs within the RAG architecture to increase accuracy and decrease the levels of hallucinations. The methods considered in this thesis are Graph Constrained Reasoning[13] (GCR), Reasoning on Graphs[12] (RoG), and Think on Graph[18] (ToG). In these approaches, an LLM is used to traverse the KG and generate paths useful for reasoning, paths that are later processed with another LLM to produce the final answer.
The goal of the thesis is to evaluate the performance of KG-based RAG approaches in situations where there is a contradiction between the parametric knowledge of LLMs and the external contextual information. As part of the goal, the research aims to compare the ability of GCR, RoG, and ToG to increase the accuracy of reasoning and avoid hallucinations. To do so, a question-answering dataset (WebQSP), with an associated KG, was altered by adding controlled perturbation and generating four new datasets, inspired by the strategy employed by Wu et al.[21]; then GCR, RoG and ToG methods were executed on the original dataset and all the modified versions, and the results of the experiment were compared.

Main contributions:

- A question-answering dataset with perturbed Knowledge Graphs, constructed starting from WebQSP and applying controlled modifications inspired by ClashEval.
- A fine-grained evaluation framework (adherent, resistant, incorrect) to analyze model behavior under contradictory contextual knowledge.
- A comparative study of the performance of three KG-based RAG methods (GCR, RoG and ToG).
- A comparison of two LLM models of different capacities (GPT-4.1-standard and GPT-4.1-nano).
- An extension of Prior/Context Bias analysis, introduced in ClashEval, to Knowledge Graph-based RAG systems.
- An analysis of failure modes, separating reasoning path generation errors from final answer generation errors.

2. Background

2.1. Large Language Models

Large Language Models (LLMs) are a type of neural network based on the Transformer architecture; transformers leverage the mechanism of attention[20] to identify dependencies between tokens in a sequence.
Text is represented as a sequence of tokens (usually words or subwords), drawn from a fixed vocabulary of tokens.
During inference, the model receives a sequence of tokens as input and generates a probability distribution over the entire vocabulary of tokens. The next token is chosen according to a decoding strategy, such as greedy search, beam search, or probabilistic sampling methods. The whole process keeps repeating until reaching a stop condition, which can be the generation of a special end-of-sequence (EOS) token, or reaching a fixed maximum length.

The training of LLMs relies on large text corpora. For a sequence of tokens, the model learns to predict the next one by minimizing the cross-entropy loss between the predicted and the true token distributions. As a result of this process, an LLM acquires the skills to generate a realistic sentence.
Thanks to the training process, the LLMs also absorb facts and contextual information from the training data. This internal (parametric) knowledge enables LLMs to respond with facts from the real world, even without directly accessing knowledge sources.
LLMs suffer from hallucinations. A hallucination occurs when the LLM produces text that is plausible, but factually incorrect or inconsistent with the input context. This happens because the model predicts tokens based on probability patterns rather than factual knowledge[6].

2.2. In-Context Learning and Shot-based Prompting

In-Context Learning (ICL) is the ability of LLMs to learn from examples and instructions embedded directly in the prompt, rather than needing additional training or fine-tuning. Shot-based prompting is the technique of adding one or more examples to the prompt to allow the model to better understand the task and expected output; it has been shown to improve LLM performance[17]. The number of examples ("shots") can vary:

- zero-shot: no examples are given and the model relies only on its parametric knowledge;
- one-shot: a single example is included;
- few-shots: two or more examples are included to show recurring patterns and achieve more accurate responses.

2.3. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a framework that combines generative language models with external knowledge retrieval. RAG consists of two modules: a retriever and a generator. The retriever identifies documents from an external knowledge base that are relevant to the query. The generator (usually an LLM) produces an output conditioned on both the input query and the retrieved content.
RAG enables the possibility of dynamic knowledge integration with the use of an external knowledge source, without the need to retrain the language model. Due to the incorporation of retrieved information from outside sources into the process of generating responses, RAG enhances information consistency, reduces hallucinations and improves transparency.

2.4. Knowledge Graphs

Knowledge Graphs (KGs) are a type of data structure used to represent structured information as a directed labeled graph, with nodes representing concrete or conceptual entities and edges representing semantic relationships between those entities.
A KG can be formally represented in the form of a collection of triples: $\mathcal{G} = \{(e, r, e') \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$, where $\mathcal{E}$ is the set of entities and $\mathcal{R}$ is the set of relations.

2.5. Relation Paths

A Relation Path is a sequence of relations: $z = \{r_1, r_2, ..., r_l\}$, where $r_i \in \mathcal{R}$ denotes the i-th relation in the path and l denotes the length of the path

2.6. Reasoning Paths

A Reasoning Path is an instance of a relation path z in KGs: $w_z = e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \cdots \xrightarrow{r_l} e_l$, where $\forall(e_{i-1}, r_i, e_i) \in \mathcal{G}$. $e_i \in \mathcal{E}$ denotes the i-th entity and r_i denotes the i-th relation in the relation path z. Multiple different instances of the same Relation Path can exist on a KG.

2.7. Prefix Tree

A prefix tree (or Trie) is a hierarchical data structure for strings where each node represents a shared prefix of its children. This makes it well-suited for encoding reasoning paths in KGs. It can be used to restrict LLM output tokens to those starting with valid prefixes[2, 3, 23].

3. Related Work

3.1. Reasoning on Graph

Reasoning on Graph (RoG) is a method developed by Luo et al.[13]. Given a question, a fine-tuned LLM is used to generate relation paths useful for answering the question; for each relation path, the matching reasoning paths are then identified on the KG and retrieved; an input containing all the retrieved reasoning paths is then submitted to a general-purpose LLM to generate the final response.
It is composed of three modules:

- planning
- retrieval
- reasoning

Planning Module

The planning module is designed to generate faithful relation paths that serve as plans for answering a given question. Leveraging the instruction-following capability of LLMs, an instruction template is used to prompt LLMs to produce relation paths. The prompt used is provided in Figure 14.
The instruction template with the question is fed to the LLM, which generates the relation paths. These paths are represented in a sentence structured as follows: $z =< PATH > r_1 < SEP > r_2 < SEP > ... < SEP > r_l < /PATH >$ where $< PATH >$, $< SEP >$, and $< /PATH >$ are special tokens representing the start, separator, and end of the relation path, respectively.

Retrieval Module

Given a question q and a relation path z, the retrieval module is responsible for obtaining the corresponding reasoning paths w_Z from the KG $\mathcal{G}$. The retrieval process involves identifying paths in $\mathcal{G}$ that originate from the entities mentioned in the question e_q and follow the structure defined by the relation path z
$\mathcal{W}_z = \{w_z(e_q, e_*) | w_z(e_q, e_*) = e_q \xrightarrow{r_1} e_1 \xrightarrow{r_2} ... \xrightarrow{r_l} e_{a*}, w_z(e_q, e_*) \in \mathcal{G}\}$
A constrained breadth-first search (BFS) is adopted to retrieve the reasoning paths w_Z from the KG.

Reasoning Module

The reasoning module receives the question q and the set of retrieved reasoning paths $\mathcal{W}_z$ and produces the final answer a. A dedicated instruction prompt is used to guide the LLM in reasoning on the paths $\mathcal{W}_z$. The reasoning paths $\mathcal{W}_z$ are formatted as a sequence of sentences structured following the template: $< entity > -> < relation > -> < entity > -> ... -> < entity >$
The reasoning module is introduced because majority voting on the retrieved reasoning paths can be affected by noise and irrelevant information, leading to incorrect answers[8, 24]. This module leverages the LLM's reasoning ability to identify the most relevant paths, discard incorrect paths, and generate more accurate results.

3.2. Graph Constrained Reasoning

Graph Constrained Reasoning (GCR) is a method developed by Luo et al.[13]. Similarly to RoG, it uses a fine-tuned LLM to generate reasoning paths grounded on a KG; after that, a general-purpose LLM is used to generate the final response.
It is composed of three modules:

- Knowledge Graph Trie Construction
- Graph-constrained Decoding
- Graph Inductive Reasoning

Knowledge Graph Trie Construction

Knowledge Graph Tries (KG-Tries) are introduced as structured indices that support reasoning grounded on the KG. A Trie can be used for representing multiple reasoning paths on a KG, and to constrain LLM token generation.
Given a KG $\mathcal{G}$ and a question q, paths W_z within L hops are retrieved from entities $e_q \in \mathcal{E}_q$ mentioned in q using a breadth-first search (BFS). Retrieved paths are formatted as sentences following the template: $< entity > -> < relation > -> < entity > -> ... -> < entity >$. Sentences are then tokenized

and stored in a KG-Trie $\mathcal{C}_\mathcal{G}$. The process can be summarized as: $\mathcal{W}_z = BFS(\mathcal{G}, \mathcal{E}_q, L)$ $\mathcal{T}_z = Tokenizer(\mathcal{W}_z)$ $\mathcal{C}_\mathcal{G} = Trie(\mathcal{T}_z)$ where $\mathcal{E}_q$ represents the question entities, L the maximum path length, and $\mathcal{T}_z$ the reasoning path tokens.
The KG-Trie $\mathcal{C}_\mathcal{G}$ serves as a decoding constraint, allowing traversal of reasoning paths in constant time $O(|\mathcal{W}_z|)$[13]. KG-Tries can be pre-built offline or generated on demand, reducing computation and latency.

Graph-constrained Decoding

The graph-constrained decoding module integrates KG-Trie with LLM generation to ensure KG-grounded reasoning.
Given a question q, an instruction prompt is submitted to the LLM to generate reasoning paths w_z and hypothesis answers a, while the KG-Trie $\mathcal{C}_\mathcal{G}$ is used to constrain decoding only to valid KG paths. Once a valid path is produced, the system switches back to normal (non-constrained) decoding to generate a hypothesis answer.
A lightweight KG-specialized LLM is fine-tuned to optimize this process, trained on triples $(q, w_z, a) \in \mathcal{D}_\mathcal{G}$ derived from KG question–answer pairs. The shortest connecting paths between question and answer entities are used as reasoning paths w_z in the training phase.
Graph-constrained decoding integrates beam search generation, enabling parallel generation of K reasoning paths and answers in a single call.

Graph Inductive Reasoning

The graph inductive reasoning module combines multiple reasoning paths and hypothesis answers to enable a general-purpose LLM to perform inductive reasoning, with the goal of producing more accurate and robust final answers.
Graph inductive reasoning aggregates the K reasoning paths and hypothesis answers generated by graph-constrained decoding, using the Fusion-in-Decoder (FiD) framework[10], to perform inductive reasoning. A general-purpose LLM is then used to generate the final answer, without the need for fine-tuning.

3.3. Think on Graph

Think on Graph (ToG) is a method introduced by Sun et al.[18]. In this framework, an LLM acts as an agent that explores a KG. Starting from an initial entity, the LLM identifies the most promising relations and neighboring entities to investigate. Following the exploration phase, the LLM determines whether it has gathered sufficient information to answer the input question; otherwise, it continues the exploration process.
The method consists of three main phases:

- Initialization
- Exploration
- Reasoning

Initialization

Given an input question, the LLM is prompted to extract the top-N entities from the KG that are mentioned in the question. The extraction may yield fewer than N entities. These topic entities serve as the starting nodes for constructing reasoning paths.

Exploration

The exploration phase is divided into two subphases: relation exploration and entity exploration.
In the relation exploration phase, all incoming and outgoing relations associated with the last entity of each reasoning path p_n are retrieved to form the set of candidate relations R_{cand} using formal graph queries.
Once the candidate relations and the corresponding extended reasoning paths P_{cand} are obtained, the LLM is used to select the top-N relations and the corresponding extended reasoning paths that are most useful for answering the question. The top-N relations are aggregated into the set R, and the top-N paths are aggregated into the set P.
It is important to note that the paths in P end with a relation, not an entity.
In the entity exploration phase, the set of candidate entities E_{cand} is created, which contains all entities reachable from the reasoning paths in P.
The reasoning paths in P are extended by appending the candidate entities, forming the set P_{cand}.
The LLM is then employed to select the top-N reasoning paths from P_{cand} that are the most relevant for

answering the question.
Compared to the beginning of the exploration phase, the new reasoning paths are extended by one hop.

Reasoning

The LLM is queried to determine whether the reasoning paths generated so far are sufficient to answer the question. If the answer is positive, the LLM is prompted to generate the final answer.
If the answer is negative, the exploration phase is repeated, followed by another reasoning phase.
If the answer is negative and the paths have reached the maximum length D_{max}, LLM is instructed to generate an answer based on its parametric knowledge.
Overall, the entire procedure requires at most $2ND + D + 1$ LLM calls.

3.4. ClashEval

Wu et al.[21] analyze the "tug-of-war" between parametric and contextual knowledge in LLMs. The authors construct a dataset of 1,200 question-answer pairs, each accompanied by a textual document containing the correct answer. The questions are categorized into five domains: Drug Dosage and Sports Record (numerical values), Dates (years), Names, and Locations. The documents associated with each question are then systematically altered. For Drug Dosage and Sports Record, a multiplicative factor is applied to the answer value (ranging from $\times 0.1$ to $\times 10.0$); for Dates, a constant is added or subtracted from the year (ranging from -100 to +100); for Names and Locations, an LLM is prompted to generate variations at three levels: Slight, Significant, and Comical.
Multiple LLMs are evaluated on this dataset. Each question is presented under three experimental conditions: without context (prior), with the correct context, and with an altered context.
The authors define Prior Bias as the probability that a model produces an incorrect answer despite being provided with the correct context, given an incorrect prior belief. This represents the likelihood that the LLM rejects correct contextual information, maintaining its own incorrect belief. They define Context Bias as the probability that the answer with the altered context is incorrect, given a correct prior. This represents the likelihood that the LLM rejects its correct belief and accepts incorrect contextual information.
Key findings include:

- The Context Bias consistently exceeds the Prior Bias.
- The model's tendency to prefer contextual information decreases as the degree of contextual alteration increases.
- Among the models analyzed, GPT-4o reports a Prior Bias of 4.1% and a Context Bias of 31.3%, while GPT-3.5 reports a Prior Bias of 5.7% and a Context Bias of 62.6%.

4. Research Questions

LLMs answer knowledge-based questions by combining two different sources of information: parametric knowledge, implicitly stored in the model weights as a result of training, and contextual knowledge, explicitly provided through the input prompt. Retrieval-Augmented Generation (RAG) systems enrich contextual knowledge by supplying externally retrieved evidence into the prompt, with the general goal of improving grounding and reducing hallucinations.
This thesis focuses on this "tug-of-war" between parametric and contextual knowledge, analyzing it in a scenario where contextual information is not provided as free text, but as knowledge paths derived from KGs using different KG-based RAG methods. These methods use an LLM to produce paths over the KG, which are then used as a structured context to support the final answer generation. Such approaches should increase factual grounding and reduce hallucinations. However, it is not guaranteed that the model will consistently follow the evidence provided by KG, especially when the evidence contradicts what the model "already knows".
To study this phenomenon, the work adopts the WebQSP question-answer dataset and its associated KGs, and introduces controlled perturbations inspired by ClashEval. By altering the answer entities, multiple dataset variants are produced with increasing levels of modification. This setup allows us to explicitly test whether KG-based RAG methods lead the model to adapt to contextual information, even when the context is progressively altered. The goal is not only to measure accuracy, but also to evaluate the model's tendency to accept or resist contextual evidence under contradiction.

The thesis addresses the following research questions:

1. How effectively do KG-based RAG methods make an LLM adhere to contextual information when it contradicts the model's parametric knowledge? How can this trade-off be quantified through Prior Bias and Context Bias?
2. How does adherence to contextual knowledge change as the degree of KG perturbation increases (Slight → Significant → Comical → Uncomp)?
3. Which method between GCR, RoG and ToG is more robust and reliable, both in terms of reasoning path generation quality and in terms of final answer accuracy?
4. When KG-based RAG produces a non-adherent answer, to what extent is the error attributable to failures in the path generation phase, versus failures in the final answer generation phase (despite the presence of correct paths)?

5. Dataset Preparation

5.1. Dataset Description

The dataset used in this work is the WebQSP[19] dataset, a benchmark designed for question answering over knowledge graphs.
Each entry in the dataset consists of a natural language question, one or more correct answers, the entities mentioned in the question that serve as starting points for reasoning, and a KG that can be exploited to infer the answer.
The Original dataset contains 1,628 entries; each entry is represented as a JSON object, and contains the following fields:

- id: a unique string identifier for the entry;
- question: the natural language question (string);
- answer: a list of strings representing all correct answers;
- q_entity: a list of entities explicitly mentioned in the question and used as the reasoning source;
- a_entity: a list of entities representing the ground-truth answers;
- graph: a list of triples of strings, where each triple describes an edge in the KG in the form ["source_entity", "relation", "target_entity"].

The graph associated with each entry encodes a local subgraph relevant to the question, containing the necessary information for reasoning from the question entities to the corresponding answers.

5.2. Dataset Cleaning

Before conducting the experiments, a data cleaning phase was performed to remove entries presenting structural inconsistencies or logical issues.
The following types of problems were identified:

- Missing question entities (q_entity) - 0 entries.
- Missing answer entities (a_entity) - 0 entries.
- Question entities not appearing in the graph - 0 entries.
- Answer entities not appearing in the graph - 71 entries (4% of the total).
- Entities present in both q_entity and a_entity - 11 entries (1% of the total).
- All answer entities unreachable from any question entities - 71 entries (4% of the total).
- Failed alteration process (see Subsection 5.3) - 90 entries (6% of the total).

It is important to note that a single entry could exhibit multiple issues among those listed above, so the total number of removed items does not correspond to the sum of the entries in type of issue.
After the cleaning phase, the final dataset was reduced to 1,462 valid entries. These entries constitute the final version of the dataset used in the experiments.

5.3. Cleaned Dataset Statistics

The cleaned dataset shows the following quantitative profile (Table 1):

	Total	Mean	Median
Number of questions	1462	-	-
Number of answer entities (a_entity)	6667	4.6	1.0
Number of question entities (q_entity)	1492	1.0	1.0
Number of graph nodes	2039468	1395.0	1531.0
Number of graph edges	5770650	3947.1	4048.5

Table 1: Table showing the quantitative profile of the cleaned WebQSP dataset. Dataset entries consist of a question, a Knowledge Graph useful for answering the question, KG entities mentioned in the question (q_entity), and KG entities considered as correct answers (a_entity). Means and medians are computed per question.

Each graph was verified to be a single connected component, ensuring reachability between each question and answer entity.
Each question was also manually categorized into one of five classes, based on the expected answer type (Table 2):

Question Class	Number of Questions
Locations	500
Names	354
Years	27
Numbers	16
Others	565

Table 2: Each question in the cleaned WebQSP dataset has been manually categorized into five categories: Locations, Names, Years, Numbers, and Others. Table shows the number of questions for each category.

The answer and a_entity fields were verified to contain exactly the same elements, ensuring consistency between the textual representations of the correct answers and their entities on the graph.

5.4. Dataset Alteration

The Original dataset was altered to produce four derivative datasets, each introducing a progressively higher degree of perturbation:

- Slight
- Significant
- Comical
- Uncomprehensible (Uncomp)

Each variant was generated by systematically modifying the answer entities (a_entity). For each entity in the answer list, a modified version was produced according to the rules described in Table 3.
The same alteration was also applied to all occurrences of the entity within the associated graph, ensuring internal consistency.
The primary goal of the Uncomp variant was to create a dataset whose answers consisted of meaningless strings, preventing any exploitation of prior knowledge by LLMs.

5.4.1 Alteration Rules by Question Class

Type	Dataset			
	Slight	**Significant**	**Comical**	**Uncomp**
Locations	LLM-based alteration	LLM-based alteration	LLM-based alteration	MD5 hash (last 8 chars)
Names	LLM-based alteration	LLM-based alteration	LLM-based alteration	MD5 hash (last 8 chars)
Years	$\pm$ 3 years	$\pm$ 30 years	$\pm$ 100 years	$\pm$ 100 years
Numbers	$\times$ 1.5	$\times$ 3	$\times$ 50	$\times$ 50
Others	LLM-based alteration	LLM-based alteration	LLM-based alteration	MD5 hash (last 8 chars)

Table 3: The original WebQSP dataset has been modified generating four progressively altered datasets: Slight, Significant, Comical and Uncomp. Each answer-entity has been modified independently, and the modified version has been propagated throughout the Knowledge Graph. Table shows the alteration rule applied to each question class. For the categories Locations, Names, and Others, an LLM has been asked to generate the altered version; for the Years category, a value shift has been applied; for the Numbers category, a multiplicative factor has been applied.

The LLM-based alteration process consists of submitting the entity to be modified to an LLM, asking it to generate three variations corresponding to the Slight, Significant, and Comical alteration levels. Each category (Names, Locations, and Others) is associated with its own specific prompt, with a single contextual example. The complete prompts used for this phase are reported in Subsection C.1. All prompts are few-shots.
For the Uncomp variant, each entity string was replaced by a hash computed using the MD5 (Message Digest 5) algorithm.
Since MD5 outputs 128 bits (32 hexadecimal characters), only the last 8 characters were retained. This shortening was chosen to simplify model processing. 8 characters were empirically verified to be sufficient to avoid hash collisions within the dataset.

6. Methods

6.1. Experimental Design

Four experiments were conducted with different methods: Plain Question, Reasoning on Graph, Graph Constrained Reasoning, and Think on Graph.
In the Plain Question (PQ) setting, each question is directly posed to the LLM without additional information. The objective of this method is to assess the prior belief of the model.
For the three KG-based methods, the path-generation and Final Answer (FA) generation phases are independently analyzed. This separation makes it possible to distinguish errors caused by missing/incorrect evidence from errors due to how the model uses the provided evidence.
The three KG-based methods are compared using multiple metrics: Adherence, Resistance, Incorrectness, Prior Bias and Context Bias.

6.2. Adherence, Resistance and Incorrectness metrics

Three metrics are introduced: Adherence measures how strongly the LLM follows the contextual knowledge provided in the prompt; Resistance measures how often the LLM rejects the modified contextual evidence and falls back to its parametric knowledge (producing the original answers); Incorrectness measures cases where the model fails entirely, producing an answer that matches neither the contextual (modified) ground truth nor the original one.

For each question, two answer sets are available: the original ground-truth answers and the modified ground-truth answers. Final predictions are compared against these sets and classified into one of three labels:

- Adherent if it matches one answer in the modified answer set.
- Resistant if it matches one answer in the original answer set.
- Incorrect otherwise.

Aggregated metrics over N questions are computed as proportions:

$$Adherence = \frac{\#adherent}{N} \quad Resistance = \frac{\#resistant}{N} \quad Incorrectness = \frac{\#incorrect}{N}$$

For the Original dataset, only the original ground-truth set is defined; nevertheless, predictions that match this ground truth are labeled adherent, because they are consistent with the contextual evidence provided in the prompt. As a consequence, on the Original dataset $Resistance = 0$ by construction.
When comparing a prediction to an answer in a given set, two matching rules are applied. StrictMatch requires exact string equality, while LooseMatch checks reciprocal substring containment (it is satisfied if the prediction contains the answer or the answer contains the prediction). More details are available in Appendix A, and the complete classification algorithm is detailed in Algorithm 1.

6.3. Prior Bias and Context Bias

Following ClashEval definitions, Prior Bias is the probability that a model ignores correct contextual information and sticks to an incorrect prior belief, while Context Bias is the probability that a model abandons a correct prior belief and follows incorrect contextual information.
In this setup, the results of Plain Question (PQ) experiments are used to assess the prior belief of the model, whereas the Final Answer (FA) denotes the model's prediction.
Prior Bias:

$$Pr(FA = incorrect \mid PQ = incorrect)$$

i.e., the system remains wrong even with correct contextual evidence.
Context Bias:

$$Pr(FA = adherent \mid PQ = correct)$$

i.e., the system follows altered contextual evidence despite having a correct prior.
To avoid mixing FA-generation failures with missing-evidence failures, evaluation of both biases is restricted to questions where at least one correct reasoning path is available.

7. Experiments Setup

The models used were GPT-4.1 (hereafter referred to as GPT-4.1-standard) and GPT-4.1-nano. Experiments using GPT-4.1-nano were performed on the full dataset, whereas those using GPT-4.1-standard were limited to a subset of 86 questions.
Each experiment was conducted twice: once asking the LLM to generate a single answer per question, and once to generate multiple answers. The analyses reported in Section 8 focus on the single-answer experiments, which allow for a more straightforward comparison; the results for the multiple-answer experiments are reported in Appendix D

7.1. Plain Question

The prompt used for PQ is provided in Figure 20. It is a one-shot prompt.
Both GPT-4.1-standard and GPT-4.1-nano were used as answer generation models.

7.2. Reasoning on Graph

RoG consists of three phases: path generation, path retrieval, and final answer generation.
The prompt used during path generation is provided in Figure 14. It is a zero-shot prompt. The model used is a fine-tuned version of LLaMA2-Chat-7B, made available by the authors of RoG at HuggingFace[1].
For each question, 3 paths are generated (by setting a dedicated parameter), as described in the original paper.
The prompt used during final answer generation is available in Figure 15. It is a zero-shot prompt. Both GPT-4.1-standard and GPT-4.1-nano were used as answer generation models.
The official implementation was released by the authors on GitHub[2].

7.3. Graph Constrained Reasoning

GCR consists of three phases: KG-Trie generation, path generation, and final answer generation.
In the KG-Trie generation phase, all paths up to 2 hops away from the topic entity are generated, following the setup described in the original paper.
The prompt used during path generation is provided in Figure 12. It is a zero-shot prompt. The model used is a fine-tuned version of LLaMA-2-7b-chat-hf, made available by the authors of GCR at HuggingFace[3].
For each question, 10 paths are generated (by setting a dedicated parameter), as in the original paper.
The prompt used during final answer generation is available in Figure 13. It is a zero-shot prompt. Both GPT-4.1-standard and GPT-4.1-nano were used as answer generation models.
The official implementation was released by the authors on GitHub[4].

7.4. Think on Graph

In ToG, the LLM acts as an active agent that iteratively explores the KG by selecting the most promising relations or entities.
The exploration parameters were set to width = 3 (a maximum of three paths explored in parallel) and depth = 3 (maximum path length of 3 hops), consistent with the original paper.
Unlike the original setup, the topic entity extraction step from the question was skipped; instead, topic entities provided in the dataset were used. This choice was made to better align ToG with RoG and GCR.
All prompts used are available in Subsection C.4 and are one-shot or few-shots. Both GPT-4.1-standard and GPT-4.1-nano were used as reasoning agents.
The official implementation was released by the authors on GitHub[5].

8. Experiments and Discussion

8.1. Think on Graph

The experiments carried out with ToG method yielded problematic results; in 77% of the experiments conducted with GPT-4.1-nano and in 50% of those with GPT-4.1-standard, the LLM failed to generate any path on the KG. An analysis of Adherence, Resistance, and Incorrectness was conducted on the few questions for which ToG was able to generate a reasoning path. Adherence on the modified datasets ranges from 1-4% for the nano model and 2-7% for the standard model.
Results are available in Appendix B.

8.2. Path Generation

During the path generation phase, a fine-tuned LLM is used to produce relation paths (for RoG) or reasoning paths (for GCR).
Reasoning paths are sequences of alternating entities and relations (i.e. $e_1 \rightarrow r_1 \rightarrow e_2 \rightarrow r_2 \rightarrow e_3$), while relation paths consist only of relations (i.e. $r_1 \rightarrow r_2 \rightarrow r_3$) and may correspond to one or more reasoning paths.
A reasoning path is classified as existing if it corresponds to a traversable path in the KG; otherwise, it is classified as non-existing. Similarly, a relation path is existing if it can be instantiated on the graph and generate at least one existing reasoning path; otherwise it is non-existing.
Ground paths are defined as the shortest paths that connect each question entity to any answer entity on the graph. A reasoning path is classified as correct if it contains at least one ground path as a contiguous subsequence, even if extended; otherwise, it is classified as incorrect. Note that a reasoning path that reaches the correct answer entity but does so without traversing any ground path is still considered incorrect. For example:

- Question: Who is the brother of Mario?
  Question entity: Mario
  Answer entity: Luigi
  Ground path: Luigi → brother_of → Mario
- The reasoning path Luigi → brother_of → Mario → work_as → plumber is considered correct (it contains the ground path, even if extended).
- The reasoning path Luigi → rival_of → Waluigi → friend_of → Wario → rival_of → Mario is considered incorrect (it does not contain any ground_path).

This choice is motivated by the fact that the terminal entity of a path is not automatically taken as the final

answer; the candidate must still pass a subsequent reasoning stage. Therefore, an extended path that embeds a ground_path can still enable the LLM to produce the correct answer.
A relation path is classified as correct if it generates at least one correct reasoning path; otherwise, it is classified as incorrect.

8.2.1 RoG

Table 4 summarizes the results of the RoG path generation phase. RoG successfully generated at least one path for all 1,462 questions in the dataset. In most cases, three paths were generated as configured; however, 38 questions resulted in empty paths, yielding fewer than three usable paths. Moreover, for 36 questions, no existing path was generated.
A significant result is that for 87% of the questions, at least one correct path was generated, making it possible, during the answer generation phase, to reach the correct answer. For the remaining 13%, the correct answer cannot be reached, as no correct path is available.
Results are highly consistent across all five datasets, since path generation depends solely on the question and graph relations, and is therefore unaffected by altered entities.

	Dataset				
	Original	**Slight**	**Significant**	**Comical**	**Uncomp**
Total number of questions	1462	1462	1462	1462	1462
Questions with less than 3 path	3%	3%	3%	3%	3%
Questions without existing paths	2%	2%	2%	2%	2%
Questions without correct paths	13%	13%	13%	13%	13%
Questions with at least 1 correct path	87%	87%	87%	87%	87%
Questions with only correct paths	18%	18%	17%	18%	18%
Total number of paths	4340	4340	4340	4340	4340
Non-existing paths	16%	16%	16%	16%	16%
Correct paths	49%	49%	49%	49%	49%

Table 4: RoG method generates up to 3 relation paths in the Knowledge Graph linking question entities to answer entities. A path is "existing" if it can be traversed in the graph, and "correct" if it exists and connects a question entity to an answer entity. Question percentages are calculated over the total number of questions, while non-existing and correct path percentages are computed over the total number of paths.

8.2.2 GCR

Table 5 reports the results of the path generation phase for GCR. For all 1,462 questions in the dataset, GCR successfully generated 10 paths, according to the configuration. In particular, constraining generation through the KG-Trie guarantees that all paths produced exist in the graph, yielding 0% invalid paths.
GCR achieves slightly higher performance than RoG, with:

- 63% correct paths (vs. 49% for RoG)
- 90% of questions containing at least one correct path (vs. 87%)
- 22% of questions with all paths correct (vs. 18%)

Unlike RoG, the results vary across the different datasets, since the path generation phase is influenced by the altered entities in the KG and not only by the relations.
As shown in Figure 1, there is a slight correlation between performance and the level of alteration of the dataset: performance decreases as the level of alteration increases, with the Comical dataset being a notable exception.

	Dataset				
	Original	**Slight**	**Significant**	**Comical**	**Uncomp**
Total number of questions	1462	1462	1462	1462	1462
Questions with less than 10 path	0%	0%	0%	0%	0%
Questions without existing paths	0%	0%	0%	0%	0%
Questions without correct paths	5%	9%	12%	7%	15%
Questions with at least 1 correct path	95%	91%	88%	93%	85%
Questions with only correct paths	23%	23%	20%	22%	20%
Total number of paths	14620	14620	14620	14620	14620
Non-existing paths	0%	0%	0%	0%	0%
Correct paths	67%	63%	60%	64%	59%

Table 5: GCR method generates up to 10 relation paths in the Knowledge Graph linking question entities to answer entities. A path is “existing” if it can be traversed in the graph, and “correct” if it exists and connects a question entity to an answer entity. Due to its internal mechanism, all the paths generated by GCR are "existing". Question percentages are calculated over the total number of questions, while non-existing and correct path percentages are computed over the total number of paths.

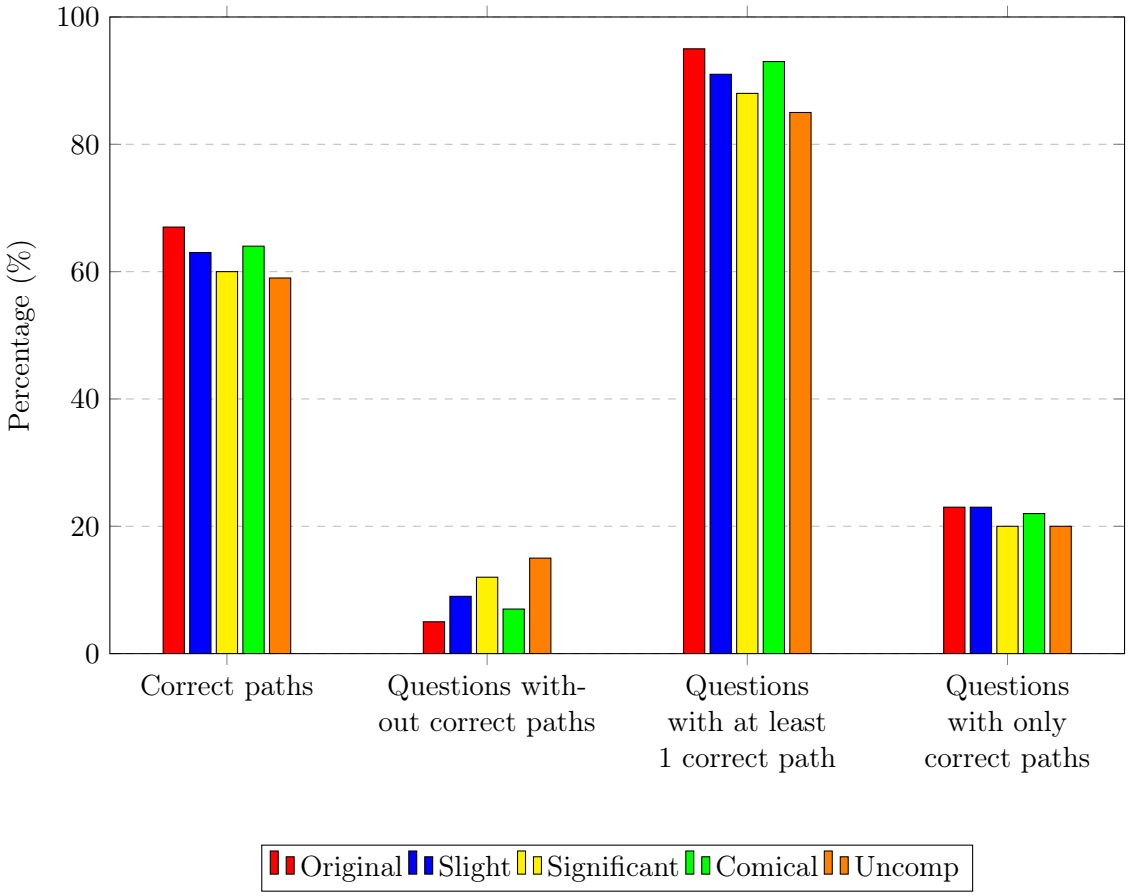

Figure 1: Bar plots showing results of the path generation phase with GCR, for all datasets. Detailed results data are reported in Table 5. Plot clearly shows how the quality of the generated paths decreases as the level of dataset alteration increases, with the Comical dataset being an exception.

8.3. Final Answer

In the Final Answer (FA) generation phase, all reasoning paths are collected and inserted, along with the question, into a dedicated prompt (Figure 13 and Figure 15). The prompt is then submitted to a general-purpose LLM, which produces the final answer.
Table 6, Figure 2 and Figure 3 report the experimental results.

Model	Method	Dataset	Adherence %		Resistance %		Incorrectness %	
nano	PQ	Original	58%	(CI: 56%-61%)	0%	(CI: 0%-0%)	42%	(CI: 39%-44%)
	GCR	Original	87%	(CI: 85%-88%)	0%	(CI: 0%-0%)	13%	(CI: 12%-15%)
		Slight	64%	(CI: 62%-67%)	10%	(CI: 8%-11%)	26%	(CI: 24%-28%)
		Significant	52%	(CI: 49%-54%)	11%	(CI: 9%-13%)	37%	(CI: 35%-40%)
		Comical	63%	(CI: 60%-65%)	10%	(CI: 8%-11%)	27%	(CI: 25%-30%)
		Uncomp	24%	(CI: 22%-26%)	21%	(CI: 19%-23%)	55%	(CI: 53%-58%)
	RoG	Original	80%	(CI: 78%-82%)	0%	(CI: 0%-0%)	20%	(CI: 18%-22%)
		Slight	50%	(CI: 47%-52%)	21%	(CI: 19%-23%)	30%	(CI: 27%-32%)
		Significant	39%	(CI: 37%-42%)	20%	(CI: 18%-22%)	41%	(CI: 38%-43%)
		Comical	42%	(CI: 40%-45%)	25%	(CI: 23%-27%)	33%	(CI: 31%-35%)
		Uncomp	7%	(CI: 6%-8%)	35%	(CI: 33%-38%)	58%	(CI: 55%-61%)
standard	PQ	Original	76%	(CI: 66%-84%)	0%	(CI: 0%-0%)	24%	(CI: 15%-34%)
	GCR	Original	88%	(CI: 81%-94%)	0%	(CI: 0%-0%)	12%	(CI: 6%-19%)
		Slight	60%	(CI: 50%-70%)	13%	(CI: 6%-20%)	27%	(CI: 17%-36%)
		Significant	57%	(CI: 47%-67%)	8%	(CI: 2%-14%)	35%	(CI: 24%-45%)
		Comical	62%	(CI: 51%-72%)	14%	(CI: 7%-22%)	24%	(CI: 16%-34%)
		Uncomp	41%	(CI: 30%-51%)	17%	(CI: 9%-26%)	42%	(CI: 31%-52%)
	RoG	Original	91%	(CI: 84%-97%)	0%	(CI: 0%-0%)	9%	(CI: 3%-16%)
		Slight	42%	(CI: 31%-52%)	30%	(CI: 21%-41%)	28%	(CI: 19%-37%)
		Significant	40%	(CI: 29%-50%)	31%	(CI: 22%-42%)	29%	(CI: 20%-38%)
		Comical	42%	(CI: 31%-52%)	30%	(CI: 21%-40%)	28%	(CI: 19%-37%)
		Uncomp	14%	(CI: 7%-22%)	57%	(CI: 47%-67%)	29%	(CI: 20%-38%)

Table 6: In the GCR and RoG methods, the question is given to an LLM together with the generated paths; in PQ, the question is presented without additional information (so it is applied only on Original dataset). Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). Table shows Adherence, Resistance and Incorrectness metrics for both models (GPT-4.1-standard and GPT-4.1-nano), all methods (PQ, GCR and RoG) and all datasets. Confidence intervals (CI) are 95% percentile bootstrap intervals computed with 10,000 resamples at the question level.

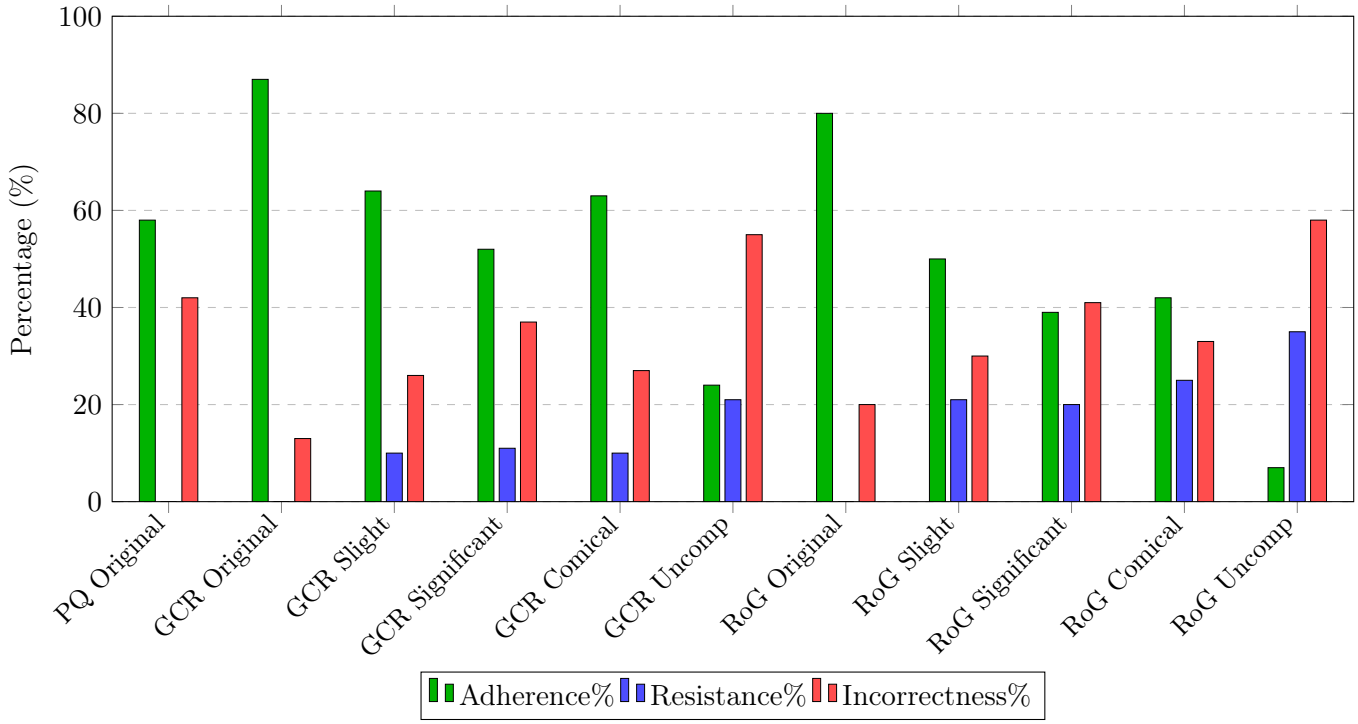

Figure 2: Bar plot showing results of the final answer generation phase with PQ, GCR and RoG using GPT-4.1-nano model, on all datasets. Detailed results data are reported in Table 6. Plot clearly shows how the performance of Adherence metric decreases as the level of dataset alteration increases, with the Comical dataset being an exception.

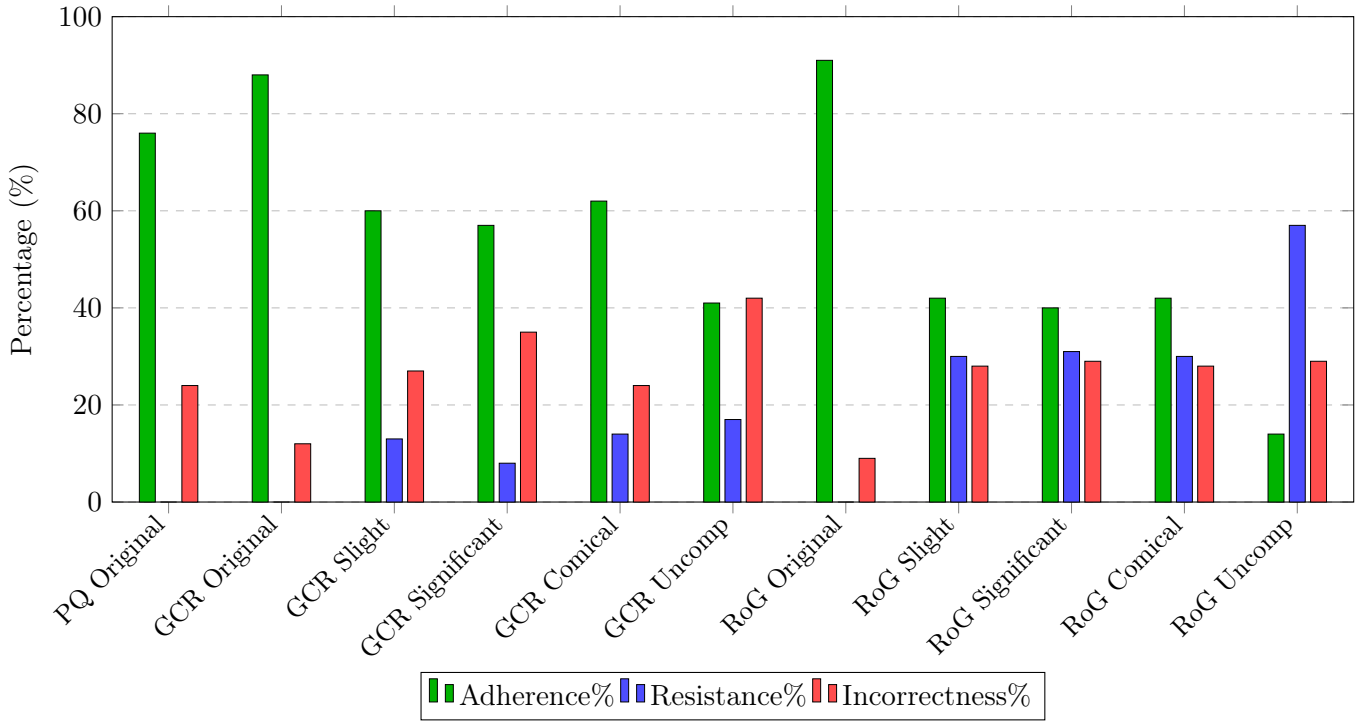

Figure 3: Bar plot showing results of the final answer generation phase with PQ, GCR and RoG using GPT-4.1-standard model, on all datasets. Detailed results data are reported in Table 6. Plot clearly shows how the performance of Adherence metric decreases as the level of dataset alteration increases, with the Comical dataset being an exception.

8.3.1 Model Comparison

Figure 4 compares the Adherence performance between the two models used, GPT-4.1-nano and GPT-4.1-standard.
Using the standard model results in an average Adherence improvement of 4%. In the experiments, both with RoG and GCR on the Slight dataset, an unexpected drop in Adherence was observed.

8.3.2 Method Comparison

Figure 5 compares the Adherence performance between the two methods, RoG and GCR.
GCR outperforms RoG, with an average Adherence increase of approximately 15%.

8.3.3 Plain Question error analysis

In the PQ experiments, performance was surprisingly low (58% accuracy for GPT-4.1-nano, 76% accuracy for GPT-4.1-standard), despite the use of a state-of-the-art model.
A manual analysis was conducted on a subset of 100 questions (out of a total of 885) that received incorrect answers.
Questions were categorized into five types of error:

- A Correct answer, but incomplete or incorrect ground truth (missing valid answers, alternate phrasings, or outdated information)
  - question: what does jamaican people speak?
  - ground truth: Jamaican English, Jamaican Creole English Language
  - prediction: Patois (Jamaican Creole)
- B Truly incorrect answer
  - question: who is niall ferguson's wife?
  - ground truth: Ayaan Hirsi Ali
  - prediction: Amanda Ferguson
- C Answer with a different granularity than the ground truth
  - question: where did eleanor roosevelt die?
  - ground truth: Manhattan
  - prediction: New York City
- D Overly general question
  - question: what was thomas jefferson role in the declaration of independence?
  - ground truth: Lawyer; Author; Writer; Statesman ...
  - prediction: Member of the Committee of Five that drafted the document
- E The model failed to provide an answer
  - question: what did peter tchaikovsky do?
  - ground truth: Somebody to Love; All Around The World; Wait for a Minute ...
  - Sorry, I cannot provide the list of songs that Justin Bieber wrote

Table 7 shows the error distribution by category.
The high proportion (44%) of category A suggests that a substantial number of answers labeled as incorrect are, in fact, valid, indicating that the true performance of GPT-4.1 may be underestimated. Therefore, all analyses that involve PQ comparisons should be interpreted with this consideration in mind.

Error Category	Number of questions
A	44
B	29
C	12
D	12
E	3
Total	100

Table 7: In the PQ method, the questions is presented to an LLM without any additional contextual information. A subset of 100 incorrect answers has been manually analyzed and grouped into five error categories (A–E). A: correct answer, but incomplete or incorrect ground truth; B: truly incorrect answer; C: answer with a different granularity than the ground truth; D: overly general question; E: model failed to provide an answer. Table shows the number of answers in each error category.

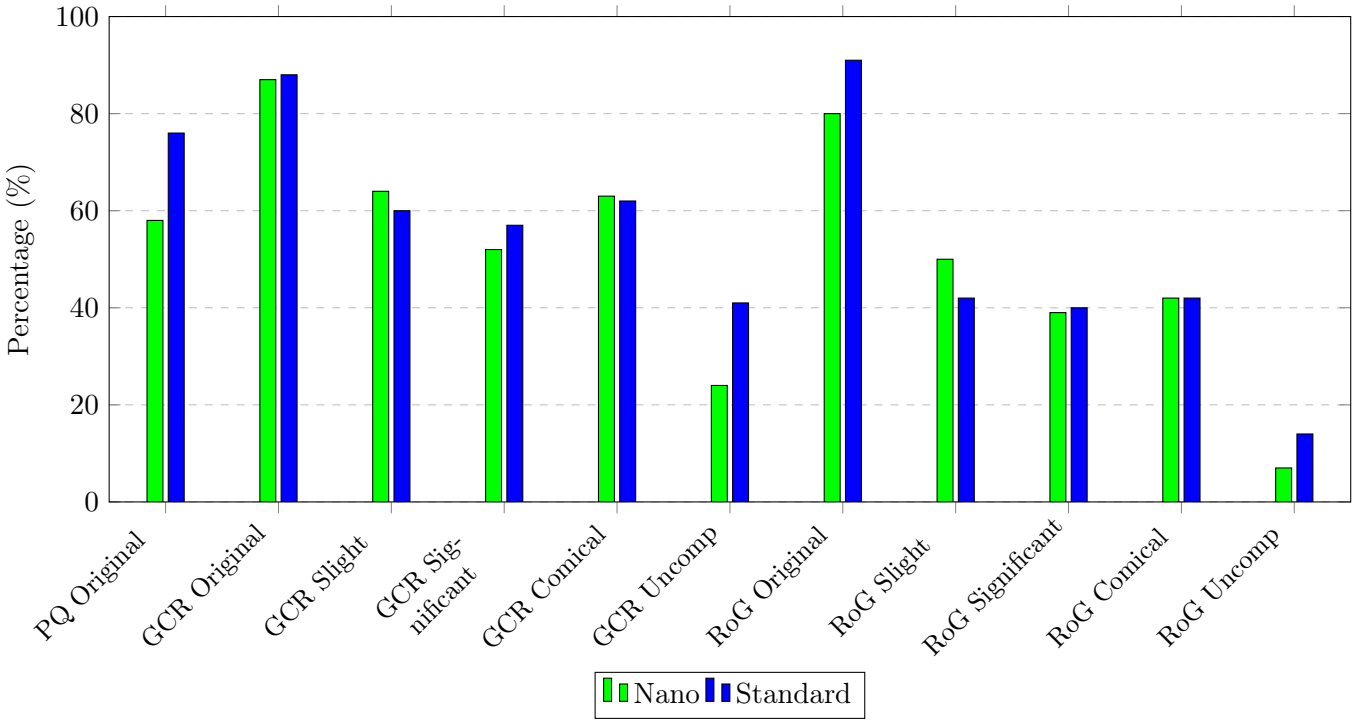

Figure 4: Bar plot showing comparison of Adherence metric between GPT-4.1-nano and GPT-4.1-standard models, on all methods (PQ, GCR and RoG) and all datasets. Detailed results data are reported in Table 6. Plot shows that the standard model, although better on average, does not always outperforms the nano model.

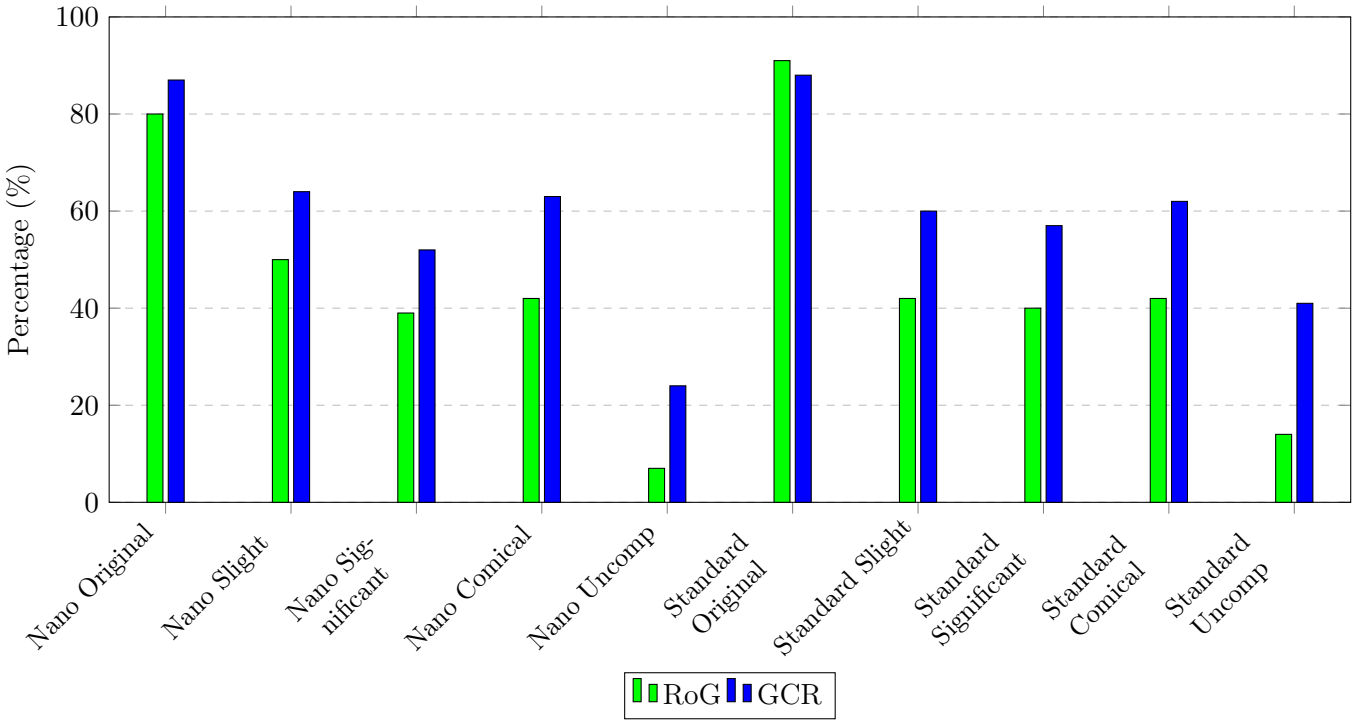

Figure 5: Bar plot showing comparison of Adherence metric between GCR and RoG methods, with both models (GPT-4.1-standard and GPT-4.1-nano) and all datasets. Detailed results data are reported in Table 6. Plot shows that GCR method always outperforms RoG, except on the Original dataset with GPT-4.1-standard model.

8.4. Comparison with the Original GCR Paper

In the original GCR paper, the evaluation metric used is Hit; a Hit occurs when the predicted answer contains at least one of the possible correct answers, and the total number of Hits is divided by the number of predictions. This metric is equivalent to Adherence, although Adherence uses a more permissive definition (allowing reciprocal inclusion).
The original paper reports a Hit value of 92.2% using GPT-4o-mini.
Table 8 summarizes the differences between the experiments in the original paper and those conducted in this thesis.
In this work, a smaller model (LLaMA-2-7B) was used for path generation, but a more powerful model (GPT-4.1) was used for the final answer generation, along with a more permissive evaluation metric.
Despite this setup, Adherence reaches 88% with GPT-4.1-standard and 87% with GPT-4.1-nano, i.e., 4 and 5 percentage points lower, respectively. Performance on altered datasets is further reduced.

	GCR original paper	**GCR thesis**
Correct answer evaluation method	Inclusion of the correct answer within the prediction (more strict)	Inclusion of the correct answer within the prediction, or vice versa (more permissive)
Model used for path generation	LLaMA-3.1-8B (fine-tuned)	LLaMA-2-7B (fine-tuned)
Model used for final answer generation	GPT-4o-mini	GPT-4.1-nano and GPT-4.1-standard
Number of predictions generated per question	multiple	single

Table 8: Table showing comparison of different implementation aspects of GCR between thesis and original paper. Compared aspects are: correct answer evaluation method, model used for path generation, model used for final answer generation and number of predictions generated per question.

8.5. Comparison with the Original RoG Paper

In the original RoG paper, the Hit metric is also used, similar to GCR.
The original paper reports a Hit rate of 81.51% using LLaMA-3.1-7B as the final model.
Table 9 summarizes the differences between the experiments in the original paper and those conducted in this thesis.
In this work, the same model (LLaMA-2-7B) was used for path generation, a more powerful model (GPT-4.1) for final answer generation, and a more lenient evaluation metric.
The Adherence value for GPT-4.1-standard on the Original dataset is 91%, and 80% for GPT-4.1-nano, both better than those reported in the original paper.
For the Slight, Significant, and Comical altered datasets, Adherence drops to approximately 40

	RoG original paper	RoG thesis
Correct answer evaluation method	Inclusion of the correct answer within the prediction (more strict)	Inclusion of the correct answer within the prediction, or vice versa (more permissive)
Model used for path generation	LLaMA-3.1-7B (fine-tuned)	LLaMA-2-7B (fine-tuned)
Model used for final answer generation	LLaMA-3.1-7B	GPT-4.1-nano and GPT-4.1-standard
Number of Predictions generated per question	single	single

Table 9: Table showing comparison of different implementation aspects of RoG between thesis and original paper. Compared aspects are: correct answer evaluation method, model used for path generation, model used for final answer generation and number of predictions generated per question.

8.6. Path–FA Cross-Analysis

To better understand the actual performance of KG-based RAG methods, and where failures originate, an analysis is conducted, for each question, to determine whether at least one correct path was generated (i.e., a path containing a ground path as a contiguous subsequence, as defined in Subsection 8.2); then the FA outcome (adherent/resistant/incorrect) is computed, conditioned on the path-availability.
This failure attribution distinguishes between two failure modes:

- Path-generation failure: the system fails upstream because no correct path is generated; in this case, the FA model cannot ground its response on KG evidence, and any adherence to the correct answer becomes unlikely or accidental.
- FA-generation failure: at least one correct path exists, but the FA model still generates a non-adherent response; this indicates that the model does not correctly use (or does not follow) the available KG evidence when producing the final answer.

Figure 6 shows the split between questions with at least one correct path (88%) and those with none (12%), followed by the FA outcomes on each branch (for the Significant dataset, using GCR and nano model). It should be noted that for 3 questions (2% of the total), the LLM produced an adherent answer despite the absence of correct paths. This occurs when some generated paths reach the correct answer entity but are not the shortest ones, and are therefore marked as incorrect; however, the presence of the target entity in the path may still guide the LLM toward the correct answer.
Table 10 quantifies the impact of path-generation failures by reporting Adherence computed on all questions (Old ADH) and Adherence computed only on questions where at least one correct path exists (New ADH); the difference measures how much non-adherence can be attributed to path-generation limitations, and the remaining non-adherence in New ADH reflects answer-generation limitations.
Table 11 reports, for each experiment, the total number of errors, defined as the sum of Resistant and Incorrect final answers, and how these errors split across the two main failure sources: missing reasoning paths vs. final answer failures.

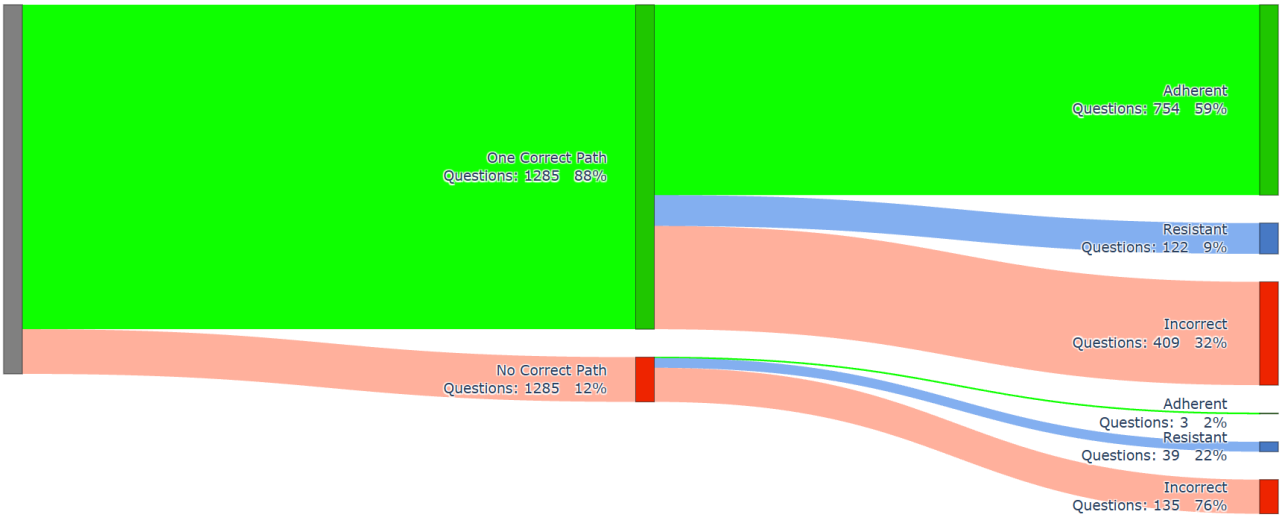

Figure 6: In the GCR and RoG methods, the question is given to an LLM together with the generated paths. Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). Sankey diagram shows the path-Final Answer cross-analysis. First split distinguishes questions with at least one correct reasoning path from those with none. Second split reports Adherence, Resistance, and Incorrectness computed on the FA results. The diagram refers to the GCR method, using the nano model, on the Significant dataset.

Model	Method	Dataset	One correct path	No correct path	Old ADH %		New ADH %		Increase
nano	GCR	Original	95%	5%	87%	(CI: 85%-88%)	90%	(CI: 89%-92%)	+3%
		Slight	91%	9%	64%	(CI: 62%-67%)	70%	(CI: 68%-72%)	+6%
		Significant	88%	12%	52%	(CI: 49%-54%)	59%	(CI: 56%-61%)	+7%
		Comical	93%	7%	63%	(CI: 60%-65%)	68%	(CI: 65%-70%)	+5%
		Uncomp	85%	15%	24%	(CI: 22%-26%)	28%	(CI: 26%-30%)	+4%
	RoG	Original	87%	13%	80%	(CI: 78%-82%)	86%	(CI: 84%-88%)	+6%
		Slight	87%	13%	50%	(CI: 47%-52%)	56%	(CI: 53%-59%)	+6%
		Significant	87%	13%	39%	(CI: 37%-42%)	45%	(CI: 42%-48%)	+6%
		Comical	87%	13%	42%	(CI: 40%-45%)	47%	(CI: 45%-50%)	+5%
		Uncomp	87%	13%	7%	(CI: 6%-8%)	8%	(CI: 6%-9%)	+1%
standard	GCR	Original	94%	6%	88%	(CI: 81%-94%)	93%	(CI: 86%-98%)	+5%
		Slight	91%	9%	60%	(CI: 50%-71%)	67%	(CI: 56%-77%)	+7%
		Significant	87%	13%	57%	(CI: 47%-67%)	65%	(CI: 55%-76%)	+8%
		Comical	94%	6%	62%	(CI: 51%-72%)	65%	(CI: 56%-75%)	+3%
		Uncomp	84%	16%	41%	(CI: 30%-51%)	49%	(CI: 38%-60%)	+8%
	RoG	Original	86%	14%	91%	(CI: 84%-97%)	96%	(CI: 91%-100%)	+5%
		Slight	84%	16%	42%	(CI: 31%-52%)	50%	(CI: 39%-61%)	+8%
		Significant	86%	14%	40%	(CI: 29%-50%)	46%	(CI: 35%-57%)	+6%
		Comical	86%	14%	42%	(CI: 31%-52%)	49%	(CI: 38%-61%)	+7%
		Uncomp	86%	14%	14%	(CI: 7%-22%)	16%	(CI: 8%-24%)	+2%

Table 10: In the GCR and RoG methods, the question is given to an LLM together with the generated paths. Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). "One correct path" is the percentage of questions with at least one correct path, while "No correct path" refers to those with none. The table compares Adherence computed only on questions with at least one correct path (New ADH) to Adherence over all questions (Old ADH). Conditioning on questions with a correct path yields only marginal improvements (2–8%). Experiments have been conducted using the GPT-4.1-standard and GPT-4.1-nano models. Confidence intervals (CI) are 95% percentile bootstrap intervals computed with 10,000 resamples at the question level.

Model	Method	Dataset	Tot Errors	Errors from path failures %	Errors from FA failures %
nano	GCR	Original	195	30%	70%
		Slight	523	24%	76%
		Significant	705	25%	75%
		Comical	544	20%	80%
		Uncomp	1110	19%	81%
	RoG	Original	291	40%	60%
		Slight	737	24%	76%
		Significant	886	21%	79%
		Comical	847	21%	79%
		Uncomp	1362	14%	86%
standard	GCR	Original	10	40%	60%
		Slight	34	24%	76%
		Significant	37	30%	70%
		Comical	33	15%	85%
		Uncomp	51	27%	73%
	RoG	Original	8	63%	38%
		Slight	50	28%	72%
		Significant	52	23%	77%
		Comical	50	24%	76%
		Uncomp	74	16%	84%

Table 11: In the GCR and RoG methods, the question is given to an LLM together with the generated paths. Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). A non-adherent answer is attributed to the path generation phase if no correct path was generated for that question; otherwise, if at least one correct path exists but the answer is still non-adherent, the error is attributed to the final answer (FA) phase. Table reports the total number of errors (Tot errors), the percentage due to path failures, and the percentage due to FA failures. Results show that most errors stem from the final answer generation phase. Experiments have been conducted using the GPT-4.1-standard and GPT-4.1-nano models.

8.7. PQ–Path–FA Cross-Analysis

Similarly, for each question, it is analyzed whether, in Plain Question mode, the model provided a correct answer, whether at least one correct path was generated, and whether the final answer was adherent, resistant, or incorrect; the results are then cross-referenced.
If the model answers correctly in PQ mode, it can be assumed that the parametric knowledge of the model contains the correct information.
The sequence PQ incorrect – One Correct Path – Adherent represents cases where the model lacks the correct answer in its parametric knowledge, but successfully follows a correct reasoning path to generate the appropriate (potentially altered) answer.
The sequence PQ correct – One Correct Path – Adherent represents cases where the model knows the correct answer, is given a path leading to an altered answer, and successfully adapts, preferring contextual over parametric knowledge.
The sequence PQ correct – One Correct Path – Resistant represents cases where the model knows the correct answer, is given a path leading to an altered answer, but fails to adapt and instead outputs the true answer, thus favoring parametric over contextual knowledge.
Table 12 reports the results of these sequences for both models, both methods, and all datasets. The Original dataset results are included for reference.

Figure 7 shows the Adherence for the PQ Correct – One Correct Path – Adherent sequence, revealing that:

- GCR outperforms RoG;
- Using a larger model (standard instead of nano) provides only marginal improvements;
  - In RoG-Slight, it even leads to a slight performance drop;
- Excluding the Uncomp dataset, Adherence values generally fall between 50% and 75%;

Model	Method	Dataset	PQ Correct One Correct path Adherence		PQ Correct One Correct Path Resistance		PQ Incorrect One Correct path Adherence	
nano	GCR	Original	95%	(CI: 93%-96%)	0%	(CI: 0%-0%)	84%	(CI: 81%-87%)
		Slight	72%	(CI: 69%-75%)	11%	(CI: 9%-14%)	67%	(CI: 63%-71%)
		Significant	59%	(CI: 55%-63%)	13%	(CI: 11%-16%)	58%	(CI: 54%-62%)
		Comical	69%	(CI: 66%-72%)	12%	(CI: 10%-14%)	66%	(CI: 62%-70%)
		Uncomp	28%	(CI: 24%-31%)	28%	(CI: 25%-31%)	28%	(CI: 25%-32%)
	RoG	Original	93%	(CI: 91%-95%)	0%	(CI: 0%-0%)	76%	(CI: 73%-80%)
		Slight	57%	(CI: 54%-61%)	26%	(CI: 23%-29%)	54%	(CI: 50%-59%)
		Significant	46%	(CI: 43%-50%)	24%	(CI: 21%-27%)	43%	(CI: 39%-47%)
		Comical	50%	(CI: 46%-54%)	31%	(CI: 28%-34%)	43%	(CI: 39%-48%)
		Uncomp	8%	(CI: 6%-10%)	50%	(CI: 47%-54%)	8%	(CI: 6%-11%)
standard	GCR	Original	98%	(CI: 95%-100%)	0%	(CI: 0%-0%)	76%	(CI: 57%-90%)
		Slight	72%	(CI: 60%-82%)	11%	(CI: 4%-19%)	52%	(CI: 33%-71%)
		Significant	76%	(CI: 65%-87%)	5%	(CI: 0%-13%)	35%	(CI: 15%-55%)
		Comical	73%	(CI: 62%-83%)	13%	(CI: 5%-23%)	43%	(CI: 24%-62%)
		Uncomp	50%	(CI: 37%-63%)	17%	(CI: 8%-29%)	45%	(CI: 25%-65%)
	RoG	Original	98%	(CI: 95%-100%)	0%	(CI: 0%-0%)	89%	(CI: 74%-100%)
		Slight	53%	(CI: 40%-66%)	30%	(CI: 19%-43%)	42%	(CI: 21%-63%)
		Significant	49%	(CI: 36%-62%)	31%	(CI: 18%-44%)	37%	(CI: 16%-58%)
		Comical	51%	(CI: 38%-64%)	29%	(CI: 18%-42%)	42%	(CI: 21%-63%)
		Uncomp	16%	(CI: 7%-27%)	67%	(CI: 55%-80%)	16%	(CI: 0%-32%)

Table 12: In the GCR and RoG methods, the question is given to an LLM together with the generated paths; in PQ, the question is presented without additional information (so it is applied only on Original dataset). Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). The table reports Adherence and Resistance computed only on questions with at least one correct path, further conditioned on PQ performance (questions answered correctly by PQ vs. those answered incorrectly by PQ). Experiments have been conducted using the GPT-4.1-standard and GPT-4.1-nano models. Confidence intervals (CI) are 95% percentile bootstrap intervals computed with 10,000 resamples at the question level.

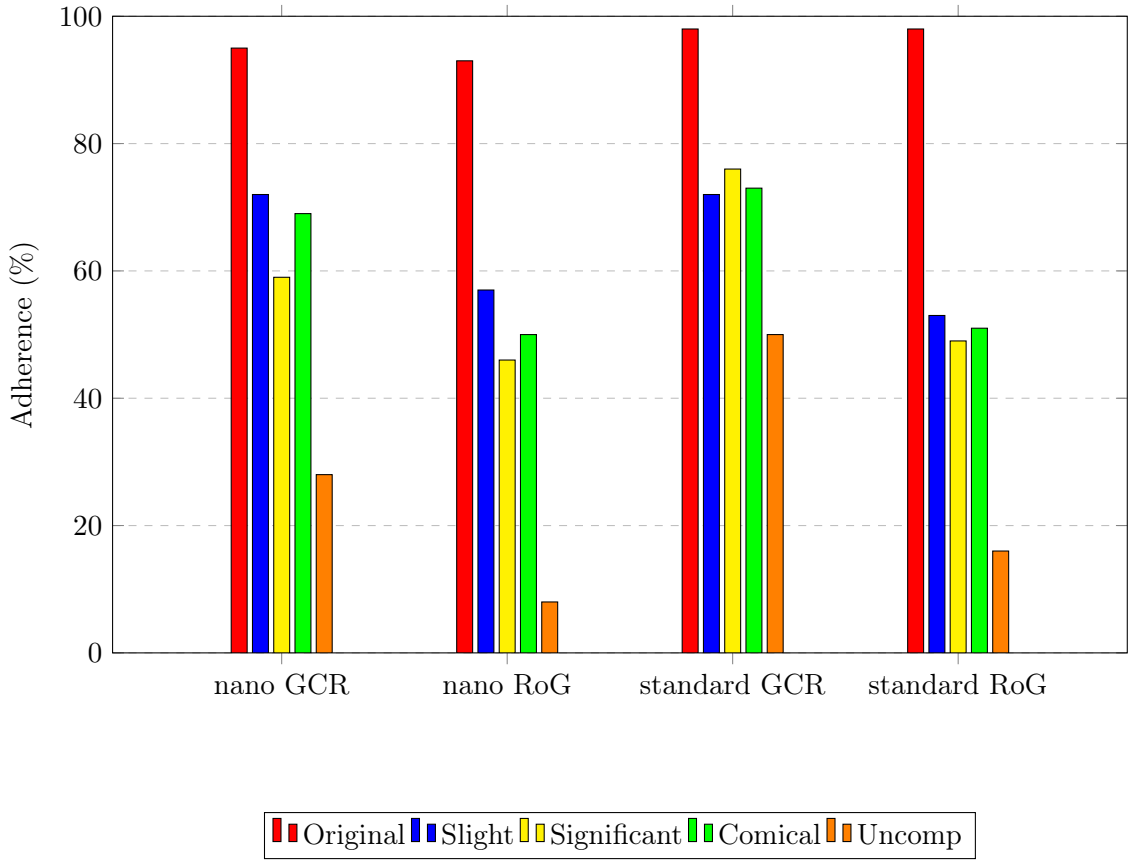

Figure 7: Bar plots showing Adherence across all datasets, for both methods (GCR and RoG) and both models (GPT-4.1-standard and GPT-4.1-nano models). Adherence is computed on questions for which the PQ method is correct and at least one valid reasoning path is generated. Detailed results data are reported in Table 12. Adherence metric is defined in Subsection 6.2. Plot shows that GCR outperforms RoG, and using a larger model (standard instead of nano) provides only marginal improvements.

8.8. Prior Bias and Context Bias

Following the approach proposed in ClashEval, both Context and Prior Biases are computed.
First, all questions for which no correct reasoning paths were generated are discarded. Then, it is analyzed whether the model in PQ (Plain Question) mode produced a correct or incorrect answer and, for both subsets, whether the answer obtained with KG support was adherent, resistant, or incorrect.
Figure 8 illustrates the splits analyzed.
According to the definition of Prior Bias, in our setting it corresponds to the sequence One Correct Path – PQ Incorrect – Incorrect, computed on the Original (unaltered) dataset. The Context Bias instead corresponds to the sequence One Correct Path – PQ Correct – Adherent, using the altered datasets.
Table 13 reports the prior and Context Bias values for both models and both methods under analysis.
In line with the results reported in ClashEval, the Context Bias decreases as the level of dataset alteration increases, except for the Comical dataset and for the Significant dataset when using the GCR method with the standard model.
The value of Context Bias reported in ClashEval is 31.3% for GPT-4o and 62.6% for GPT-3.5. The observed Context Bias for RoG (for both models and across all datasets, excluding Uncomp) falls between the two values previously reported, while the Context Bias for GCR is higher than both; one of the causes could be the fact that the prompts used (Figure 13 and Figure 15) explicitly instruct the model to adapt to the contextual information provided.
In contrast, the Prior Bias (for both models and methods) is substantially higher than that reported in ClashEval (4.1% for GPT-4o and 5.7% for GPT-3.5).

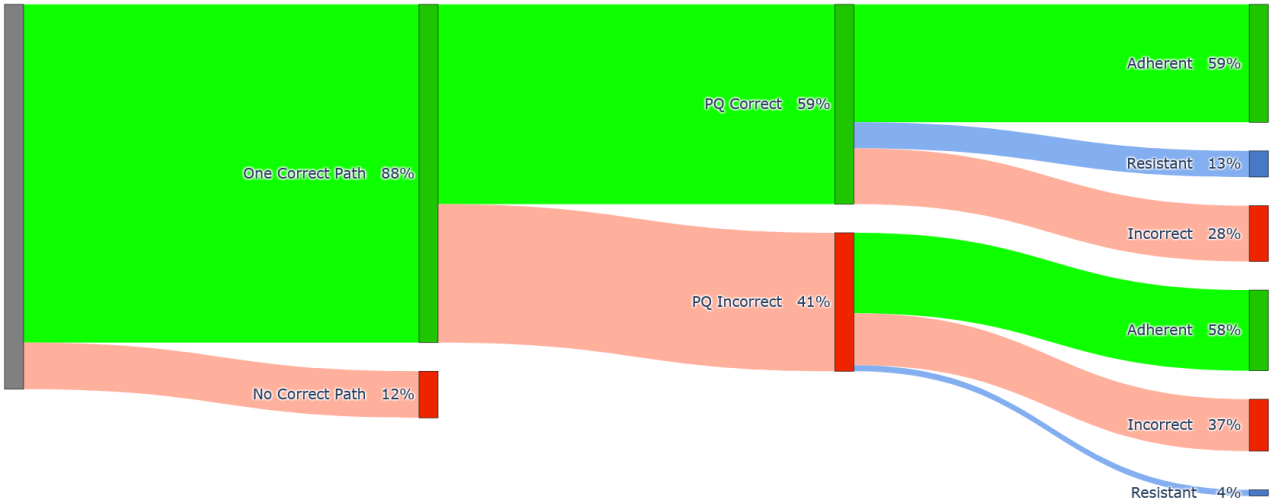

Figure 8: In the GCR and RoG methods, the question is given to an LLM together with the generated paths; in PQ, the question is presented without additional information (so it is applied only on Original dataset). Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). Sankey diagram shows the path–PQ–FA cross-analysis. The first split separates questions with at least one correct reasoning path from those with none. The second split distinguishes questions correctly answered by PQ from those answered incorrectly. The third split shows Adherence, Resistance, and Incorrectness based on the FA results. The diagram refers to the GCR method, using the nano model, on the Significant dataset.

Model	**Method**		**Dataset**				
			Original	**Slight**	**Significant**	**Comical**	**Uncomp**
nano	GCR	Prior Bias	16%	-	-	-	-
		Context Bias	-	72%	59%	69%	28%
	RoG	Prior Bias	24%	-	-	-	-
		Context Bias	-	57%	46%	50%	8%
standard	GCR	Prior Bias	24%	-	-	-	-
		Context Bias	-	72%	76%	73%	50%
	RoG	Prior Bias	11%	-	-	-	-
		Context Bias	-	53%	49%	51%	16%

Table 13: In the GCR and RoG methods, the question is given to an LLM together with the generated paths; in PQ, the question is presented without additional information, to assert the model's prior knowledge. Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). Prior Bias (computed only on Original dataset) is the probability that a model ignores correct context and follows an incorrect prior belief; Context Bias (computed only on altered datasets) is the probability that it abandons a correct prior and follows incorrect context (see Subsection 6.3 for details). Table reports Prior Bias and Context Bias for both methods (GCR and RoG), across both models (GPT-4.1-standard and GPT-4.1-nano) and all datasets. Results show that the Prior Bias is non-negligible, whereas the Context Bias generally decreases as the level of dataset alteration increases.

8.9. Discussion

This section answers the research questions using the experimental evidence reported in the previous sections.

RQ1.1 - Adherence to contradictory contextual knowledge Under contradiction (altered datasets), Adherence is substantial but not guaranteed: even when at least one correct path exists, Adherence typically falls in the ∼50-75% band (excluding the Uncomp variant), as reported in Table 6, showing that the model does not always follow KG evidence when it conflicts with parametric knowledge.

RQ1.2 - Prior Bias and Context Bias The bias analysis shows that Context Bias is generally high but decreases with stronger perturbations (nano GCR: 72% → 59% → 69% → 28%; nano RoG: 57% → 46% → 50% → 8%), again with an exception on the Comical dataset. Prior Bias is non-negligible on the Original dataset, as shown in Table 13, indicating that even with correct paths available, the model can still fail to override incorrect priors.

RQ2 - Effect of increasing KG perturbation For both RoG and GCR, Adherence generally decreases as perturbation grows from Slight to Significant, and collapses on Uncomp (e.g., nano: GCR 64% → 52% → 24%; RoG 50% → 39% → 7%), with the Comical variant being a consistent exception. The decrease is shown in Figure 2 and Figure 3, and the complete results presented in Table 6. Overall, stronger perturbations reduce the model's ability to align with contextual knowledge, especially when the context becomes semantically unintelligible.

RQ3 - Most robust method ToG is not robust in this setting: it fails to generate any path in a large fraction of questions (∼77% with nano; ∼50% with standard), and has very low Adherence value, as shown in Table 14 and Table 15. Between the remaining methods, GCR is the most reliable: it produces only valid paths by construction (0% non-existing paths) and delivers higher path correctness (∼63% correct paths for GCR vs ∼49% for RoG), as reported in Table 4 and Table 5; also, GCR has higher final Adherence than RoG (+15% on average), as shown in Figure 5. RoG remains competitive on the Original dataset (notably 91% with the standard model), but degrades more steeply under perturbation.

RQ4 - Failure attribution Failures, with both RoG and GCR, are for the most part attributable to the final answer generation (FA) phase, not to path generation; percentages range from 60% to 86% of errors attributable to failures in FA, across all experiments (with the exception of RoG with standard model on Original dataset), as reported in Table 11.
When restricting evaluation to questions with at least one correct path (New ADH), Adherence improves only by a few percentage points compared to Old ADH (typically +3%-8% across all experiments), as reported in Table 10.
This means that a substantial portion of errors remain even when correct evidence is available, indicating that the main bottleneck is the model's ability to correctly use the generated paths.

9. Conclusions

In this work, an analysis was conducted on the ability of LLMs to adhere to in-prompt information when presented through RAG-like strategies based on reasoning over KGs, even when such information is incorrect and contradicts the parametric knowledge of the model. Starting from the WebQSP dataset, four variants (Slight, Significant, Comical, and Uncomp altered) were produced, following an approach similar to that proposed in ClashEval. Each variant was obtained by modifying the correct answers and applying the same modifications to entities in the KG.
The experimental analysis involved three KG-based methods (ToG, RoG, and GCR), along with a Plain Question (PQ) baseline without KG support. The tests were conducted mainly with GPT-4.1-nano, while a subset of the questions was also tested with GPT-4.1-standard.
The results show clear differences between the approaches. ToG proved to be unreliable: in only about half of the cases it was able to generate at least one usable reasoning path, limiting its capacity to be evaluated. RoG

showed lower performance than the one reported in the original publication, while GCR achieved better-than-expected results, emerging as the most effective method.
The analysis of Prior Bias and Context Bias reveals the presence of a certain degree of alignment with the modified KG information, although this effect is not absolute. The observed Resistance metric (ranging between 10% and 20% depending on the dataset variant) indicates that LLMs do not automatically conform to the knowledge presented, even when the prompt explicitly instructs them to do so.
This work presents limitations:

- The dataset questions are open-ended rather than multiple-choice, making automatic accuracy evaluation more difficult. This particularly affects the reliability of the PQ baseline.
- Tests with the more powerful model, GPT-4.1-standard, were conducted on only a subset of 86 questions due to usage costs, preventing a comprehensive large-scale comparison.
- Although three reasoning-path-based methods were initially planned (ToG, RoG, and GCR), ToG was excluded from the analyses due to its high failure rate in path generation. Because of that, the comparative analysis includes only two functioning methods: RoG and GCR, reducing the breadth of the experimental comparison.

Based on the evidence collected, multiple directions for further investigation are possible:

- An in-depth analysis of ToG failures to understand the causes of the low percentage of valid reasoning paths and to evaluate potential fixes.
- Evaluation using different models, including LLMs of various sizes and architectures, to verify the robustness of the observed data.
- Testing on alternative datasets. The CWQ (Complex Web Questions) dataset was considered but excluded because path generation is significantly slower due to the complexity of reasoning chains. Other domain-specific datasets could help assess the generalizability of the conclusions to different types of knowledge.

Notes

[1]https://huggingface.co/rmanluo/RoG
[2]https://github.com/RManLuo/reasoning-on-graphs
[3]https://huggingface.co/rmanluo/GCR-LLaMA-2-7b-chat-hf
[4]https://github.com/RManLuo/graph-constrained-reasoning
[5]https://github.com/DataArcTech/ToG

References

[1] Muhammad Aurangzeb Ahmad, Ilker Yaramis, and Taposh Dutta Roy. Creating trustworthy llms: Dealing with hallucinations in healthcare ai, 2023.

[2] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval, 2021.

[3] Chen Chen, Yufei Wang, Bing Li, and Kwok-Yan Lam. Knowledge is flat: A seq2seq generative framework for various knowledge graph completion, 2022.

[4] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation, 2023.

[5] Debadutta Dash, Rahul Thapa, Juan M. Banda, Akshay Swaminathan, Morgan Cheatham, Mehr Kashyap, Nikesh Kotecha, Jonathan H. Chen, Saurabh Gombar, Lance Downing, Rachel Pedreira, Ethan Goh, Angel Arnaout, Garret Kenn Morris, Honor Magon, Matthew P Lungren, Eric Horvitz, and Nigam H. Shah. Evaluation of gpt-3.5 and gpt-4 for supporting real-world information needs in healthcare delivery, 2023.

[6] Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. On the origin of hallucinations in conversational models: Is it the datasets or the models?, 2022.

[7] Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension, 2019.

[8] Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Improving multi-hop knowledge base question answering by learning intermediate supervision signals, 2021.

[9] Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie

Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. Trustllm: Trustworthiness in large language models, 2024.

[10] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering, 2021.

[11] Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering, 2022.

[12] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning, 2024.

[13] Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models, 2025.

[14] Anthony J. Nastasi, Katherine R. Courtright, Scott D. Halpern, and Gary E. Weissman. Does chatgpt provide appropriate and equitable medical advice?: A vignette-based, clinical evaluation across care contexts. *medRxiv*, 2023.

[15] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Med-halt: Medical domain hallucination test for large language models, 2023.

[16] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019.

[17] Rafael Valencia-García Ronghao Pan, José Antonio García-Díaz. Comparing fine-tuning, zero and few-shot strategies with large language models in hate speech detection in english. *Computer Modeling in Engineering & Sciences*, 140(3):2849–2868, 2024.

[18] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph, 2024.

[19] Wen tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Annual Meeting of the Association for Computational Linguistics*, 2016.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[21] Kevin Wu, Eric Wu, and James Zou. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence, 2025.

[22] Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts, 2024.

[23] Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosha Chen, and Huajun Chen. From discrimination to generation: Knowledge graph completion with generative transformer, 2023.

[24] Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. Subgraph retrieval enhanced model for multi-hop knowledge base question answering, 2022.

A. Correctness evaluation method

In the original GCR and RoG papers, an answer generated by the LLM p is considered correct if, given a set of ground truth answers A, p contains at least one string $a \in A$ as a substring: $isCorrect(p, A) := \exists\ a \in A \mid p\ contains\ a$ This evaluation method allows small phrasing variations to still be considered correct. For example, given $A = \{"Gregory\ House"\}$ and $p =" Dr.\ Gregory\ House"$, the prediction is considered correct. In this work, the evaluation system was expanded to include reciprocal inclusion: $isCorrect(p, A) := \exists\ a \in A \mid p\ contains\ a\ \vee\ a\ contains\ p$ This evaluation scheme applies to determining the correctness for PQ, GCR, and RoG on the Original dataset. However, for altered datasets, GCR and RoG answers can be classified as adherent, resistant, or incorrect using Algorithm 1.

Algorithm 1 Classify Prediction Algorithm

```
function StrictMatch(str1, str2)
    return (str1 = str2)
end function

function LooseMatch(str1, str2)
    return (str1 ∈ str2) ∨ (str2 ∈ str1)
end function

function ClassifyPrediction(prediction, original_answers, modified_answers)
    for all ans ∈ modified_answers do
        if StrictMatch(prediction, ans) then
            return "adherent"
        end if
    end for
    for all ans ∈ original_answers do
        if StrictMatch(prediction, ans) then
            return "resistant"
        end if
    end for
    for all ans ∈ modified_answers do
        if LooseMatch(prediction, ans) then
            return "adherent"
        end if
    end for
    for all ans ∈ original_answers do
        if LooseMatch(prediction, ans) then
            return "resistant"
        end if
    end for
    return "incorrect"
end function
```

B. ToG Results

This section reports the results of the experiments conducted with ToG.
Table 14 reports the number of questions for which ToG was unable to generate any reasoning path.
Table 15 reports the Adherence, Resistance, and Incorrectness metrics computed only on the questions for which ToG was able to generate a reasoning path.

Model		Dataset				
		Original	Slight	Significant	Comical	Uncomp
nano	Total answers	1462	1462	1462	1462	1462
	Number of answers with no paths	1124	1128	1133	1113	1151
	% of answers with no paths	77%	77%	77%	76%	79%
standard	Total answers	86	86	86	86	86
	Number of answers with no paths	43	38	43	44	42
	% of answers with no paths	50%	44%	50%	51%	49%

Table 14: In ToG, an LLM acts as an agent that explores the Knowledge Graph, generating paths and then producing the final answer. Table reports the number of questions (absolute and percentage) for which ToG failed to generate any reasoning path. Results show how ToG proved to be unreliable, failing to generate paths in a large number of cases. Experiments have been conducted using the GPT-4.1-standard and GPT-4.1-nano models.

Model	Dataset	Number of questions	Questions with at least one path	Adherence %	Resistance %	Incorrectness %
nano	Original	1462	23%	29%	0%	71%
	Slight	1462	24%	2%	33%	65%
	Significant	1462	23%	1%	29%	71%
	Comical	1462	23%	4%	33%	63%
	Uncomp	1462	21%	0%	47%	53%
standard	Original	86	50%	45%	0%	55%
	Slight	86	49%	2%	49%	48%
	Significant	86	50%	2%	42%	56%
	Comical	86	56%	7%	38%	55%
	Uncomp	86	51%	0%	40%	60%

Table 15: In ToG, an LLM acts as an agent that explores the Knowledge Graph, generating paths and then producing the final answer. Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). The table reports the total number of questions, the percentage for which ToG generated a reasoning path, and Adherence, Resistance, and Incorrectness computed only on those questions. Results show that adherence is extremely low, even in cases where ToG succeeds in generating a path. Experiments have been conducted using the GPT-4.1-standard and GPT-4.1-nano models.

C. Prompts

C.1. Dataset Alteration

Figure 9 shows the prompt used to alter the "names" dataset category.
Figure 10 shows the prompt used to alter the "locations" dataset category.
Figure 11 shows the prompt used to alter the "others" dataset category.

C.2. Graph Constrained Reasoning

Figure 12 shows the prompt used to generate reasoning paths in Graph Constrained Reasoning.
Figure 13 shows the prompt used to generate the final answer in Graph Constrained Reasoning.

C.3. Reasoning on Graph

Figure 14 shows the prompt used to generate relation paths in Reasoning on Graph.
Figure 15 shows the prompt used to generate the final answer in Reasoning on Graph.

C.4. Think on Graph

Figure 16 shows the prompt used to score entity relevance in Think on Graph.
Figure 17 shows the prompt used to evaluate if enough knowledge has been retrieved to answer the question in Think on Graph.
Figure 18 shows the prompt used to generate the final answer, conditioned on the reasoning paths, in Think on Graph.
Figure 19 shows the prompt used to generate the final answer, without reasoning paths, in Think on Graph.

C.5. Plain Question

Figure 20 shows the prompt used to generate the final answer in Plain Question.

```
Your job is to modify a given name in three different ways.
In the first modification, make a slight change according to the following guidelines:
- The modified name should stay within the same country of origin, time period, and/or
gender.
In the second modification, make a significant change according to the following
guidelines:
- The modified name be from a different country of origin, time period, and/or gender.
- The modified name should be a real person's name of similar importance, stature, and
popularity.
In the third modification, come up with a comical variation of the name.  (Something in
the spirit of Boaty McBoatFace).

The modified name should have a first AND last name.

Return a json where the first key is "slight" for the slightly changed name and the
second key is "significant" for the significantly changed name, and the third key is
"comical" for a comical variation on the name.

Respond only with the JSON output, and nothing else.
Do not include any additional text or explanation.
Do not include the original name in the output.
Do not include any code delimiters.

Example Input Format:
Name:  Benjamin Franklin

Example Output:
"slight":  "Thomas Washington", "significant":  "Yi Zhou", "comical":  "Benjamjam
Franklerford"

The name to be modified is:  <name>
```

Figure 9: Prompt used to alter the "names" dataset category.

```
Your job is to modify a given name in three different ways.
In the first modification, make a slight change according to the following guidelines:
- The modified name should stay within the same country of origin, time period, and/or
gender.
In the second modification, make a significant change according to the following
guidelines:
- The modified name be from a different country of origin, time period, and/or gender.
- The modified name should be a real person's name of similar importance, stature, and
popularity.
In the third modification, come up with a comical variation of the name.  (Something in
the spirit of Boaty McBoatFace).

The modified name should have a first AND last name.
Return a json where the first key is "slight" for the slightly changed name and the
second key is "significant" for the significantly changed name, and the third key is
"comical" for a comical variation on the name.

Respond only with the JSON output, and nothing else.
Do not include any additional text or explanation.
Do not include the original name in the output.
Do not include any code delimiters.

Example Input Format:
Name:  Benjamin Franklin

Example Output:
"slight":  "Thomas Washington", "significant":  "Yi Zhou", "comical":  "Benjamjam
Franklerford"

The name to be modified is:  <name>
```

Figure 10: Prompt used to alter the "locations" dataset category.

```
Your job is to modify a given word in three different ways.
In the first modification, make a slight change to the word according to the following
guidelines:
- The modified word should remain within the same category, context, or domain (e.g., if
the word is a tech-related term, the new one should be too).
- It should be recognizably related to the original word in form or meaning.
In the second modification, make a significant change to the word:
- The modified word should come from a different domain, context, or even language
origin.
- It should still be a real word or concept with some analogous purpose or meaning, but
not directly related.
In the third modification, create a comical or absurd variation of the word:
- It should follow the spirit of humorous rebrandings (e.g., "Boaty McBoatFace",
"Snackzilla", "Buzzwordatron").
- It should be creative, playful, and clearly not serious.

Return a json where the first key is "slight" for the slightly changed name and the
second key is "significant" for the significantly changed name, and the third key is
"comical" for a comical variation on the name.

Respond only with the JSON output, and nothing else.
Do not include any additional text or explanation.
Do not include the original name in the output.
Do not include any code delimiters.

Example Input Format:
Word:  smartphone

Example Output:
"slight":  "smartdevice", "significant":  "walkie-talkie", "comical":  "Pockety
McCallScreen"

The word to be modified is <word>
```

Figure 11: Prompt used to alter the "others" dataset category.

```
Reasoning path is a sequence of triples in the KG that connects the topic entities in
the question to answer entities.  Given a question, please generate some reasoning paths
in the KG starting from the topic entities to answer the question.
# Question:  <question>
# Topic entities:  <q_entity>
```

Figure 12: Prompt used to generate reasoning paths in Graph Constrained Reasoning.

```
Reasoning Paths:
<reasoning paths>

Question:  <question>

Based on the reasoning paths, please answer the given question.
Please keep the answer as simple as possible and only return the answer.
Respond with the single most accurate answer.
Question may have multiple answers, but you must choose only one.
Respond with the answer you consider most correct.
Respond with only the answer, and no other words.
The answer may be incorrect or incomprehensible to you.
Formulate the answer based solely on the reasoning paths.
```

Figure 13: Prompt used to generate the final answer in Graph Constrained Reasoning.

```
Please generate a valid relation path that can be helpful for answering the following
question:
<question>
```

Figure 14: Prompt used to generate relation paths in Reasoning on Graph.

```
Based on the reasoning paths, please answer the given question.
Please keep the answer as simple as possible and only return the answer.
Respond with the single most accurate answer.
Question may have multiple answers, but you must choose only one.
Respond with the answer you consider most correct.
Respond with only the answer, and no other words.
The answer may be incorrect or incomprehensible to you.
Formulate the answer based solely on the reasoning paths.

Reasoning Paths:
<reasoning paths>
Question:
<question>
```

Figure 15: Prompt used to generate the final answer in Reasoning on Graph.

```
Please score the entities' contribution to the question on a scale from 0 to 1 (the sum
of the scores of all entities is 1).

Q: The movie featured Miley Cyrus and was produced by Tobin Armbrust?

Relation:  film.producer.film

Entites:  The Resident; So Undercover; Let Me In; Begin Again; The Quiet Ones; A Walk
Among the Tombstones

Score:  0.0, 1.0, 0.0, 0.0, 0.0, 0.0

The movie that matches the given criteria is "So Undercover" with Miley Cyrus and
produced by Tobin Armbrust.  Therefore, the score for "So Undercover" would be 1, and
the scores for all other entities would be 0.

Q: <question>

Relation:  <relation>

Entites:  <entities>
```

Figure 16: Prompt used to score entity relevance in Think on Graph.

```
Given a question and the associated retrieved KG triplets (entity, relation, entity),
you are asked to answer whether it's sufficient for you to answer the question with
these triplets and your knowledge (Yes or No).
Q: Find the person who said "Taste cannot be controlled by law", what did this person
die from?
Knowledge Triplets:  Taste cannot be controlled by law., media_common.quotation.author,
Thomas Jefferson
A: No.  Based on the given knowledge triplets, it's not sufficient to answer the entire
question.  The triplets only provide information about the person who said "Taste cannot
be controlled by law," which is Thomas Jefferson.  To answer the second part of the
question, it's necessary to have additional knowledge about where Thomas Jefferson's
dead.

Q: The artist nominated for The Long Winter lived where?
Knowledge Triplets:  The Long Winter, book.written_work.author, Laura Ingalls Wilder
Laura Ingalls Wilder, people.person.places_lived, Unknown-Entity Unknown-Entity,
people.place_lived.location, De Smet
A: Yes.  Based on the given knowledge triplets, the author of The Long Winter, Laura
Ingalls Wilder, lived in De Smet.  Therefore, the answer to the question is De Smet.

Q: Who is the coach of the team owned by Steve Bisciotti?
Knowledge Triplets:  Steve Bisciotti, sports.professional_sports_team.owner_s, Baltimore
Ravens Steve Bisciotti, sports.sports_team_owner.teams_owned, Baltimore Ravens Steve
Bisciotti, organization.organization_founder.organizations_founded, Allegis Group
A: No.  Based on the given knowledge triplets, the coach of the team owned by Steve
Bisciotti is not explicitly mentioned.  However, it can be inferred that the team owned
by Steve Bisciotti is the Baltimore Ravens, a professional sports team.  Therefore,
additional knowledge about the current coach of the Baltimore Ravens can be used to
answer the question.

Q: Rift Valley Province is located in a nation that uses which form of currency?
Knowledge Triplets:  Rift Valley Province, location.administrative_division.country,
Kenya Rift Valley Province, location.location.geolocation, UnName_Entity Rift Valley
Province, location.mailing_address.state_province_region, UnName_Entity Kenya,
location.country.currency_used, Kenyan shilling
A: Yes.  Based on the given knowledge triplets, Rift Valley Province is located in
Kenya, which uses the Kenyan shilling as its currency.  Therefore, the answer to the
question is Kenyan shilling.

Q: The country with the National Anthem of Bolivia borders which nations?
Knowledge Triplets:  National Anthem of Bolivia,
government.national_anthem_of_a_country.anthem, UnName_Entity National Anthem
of Bolivia, music.composition.composer, Leopoldo Benedetto Vincenti National
Anthem of Bolivia, music.composition.lyricist, José Ignacio de Sanjinés
UnName_Entity, government.national_anthem_of_a_country.country, Bolivia Bolivia,
location.country.national_anthem, UnName_Entity
A: No.  Based on the given knowledge triplets, we can infer that the National Anthem
of Bolivia is the anthem of Bolivia.  Therefore, the country with the National Anthem
of Bolivia is Bolivia itself.  However, the given knowledge triplets do not provide
information about which nations border Bolivia.  To answer this question, we need
additional knowledge about the geography of Bolivia and its neighboring countries.

Q: <question>
```

Figure 17: Prompt used to evaluate if enough knowledge has been retrived to answer the question in Think on Graph.

```
Given a question and the associated retrieved KG triplets (entity, relation, entity),
you are asked to answer the question with these triplets and your knowledge.
Q: Find the person who said "Taste cannot be controlled by law", what did this person
die from?
Knowledge Triplets:  Taste cannot be controlled by law., media_common.quotation.author,
Thomas Jefferson
A: Based on the given knowledge triplets, it's not sufficient to answer the entire
question.  The triplets only provide information about the person who said "Taste cannot
be controlled by law," which is Thomas Jefferson.  To answer the second part of the
question, it's necessary to have additional knowledge about where Thomas Jefferson's
dead.

Q: The artist nominated for The Long Winter lived where?
Knowledge Triplets:  The Long Winter, book.written_work.author, Laura Ingalls Wilder
Laura Ingalls Wilder, people.person.places_lived, Unknown-Entity Unknown-Entity,
people.place_lived.location, De Smet
A: Based on the given knowledge triplets, the author of The Long Winter, Laura Ingalls
Wilder, lived in De Smet.  Therefore, the answer to the question is De Smet.

Q: Who is the coach of the team owned by Steve Bisciotti?
Knowledge Triplets:  Steve Bisciotti, sports.professional_sports_team.owner_s, Baltimore
Ravens Steve Bisciotti, sports.sports_team_owner.teams_owned, Baltimore Ravens Steve
Bisciotti, organization.organization_founder.organizations_founded, Allegis Group
A: Based on the given knowledge triplets, the coach of the team owned by Steve Bisciotti
is not explicitly mentioned.  However, it can be inferred that the team owned by Steve
Bisciotti is the Baltimore Ravens, a professional sports team.  Therefore, additional
knowledge about the current coach of the Baltimore Ravens can be used to answer the
question.

Q: Rift Valley Province is located in a nation that uses which form of currency?
Knowledge Triplets:  Rift Valley Province, location.administrative_division.country,
Kenya Rift Valley Province, location.location.geolocation, UnName_Entity Rift Valley
Province, location.mailing_address.state_province_region, UnName_Entity Kenya,
location.country.currency_used, Kenyan shilling
A: Based on the given knowledge triplets, Rift Valley Province is located in Kenya,
which uses the Kenyan shilling as its currency.  Therefore, the answer to the question
is Kenyan shilling.

Q: The country with the National Anthem of Bolivia borders which nations?
Knowledge Triplets:  National Anthem of Bolivia,
government.national_anthem_of_a_country.anthem, UnName_Entity National Anthem
of Bolivia, music.composition.composer, Leopoldo Benedetto Vincenti National
Anthem of Bolivia, music.composition.lyricist, José Ignacio de Sanjinés
UnName_Entity, government.national_anthem_of_a_country.country, Bolivia Bolivia,
location.country.national_anthem, UnName_Entity
A: Based on the given knowledge triplets, we can infer that the National Anthem of
Bolivia is the anthem of Bolivia.  Therefore, the country with the National Anthem
of Bolivia is Bolivia itself.  However, the given knowledge triplets do not provide
information about which nations border Bolivia.  To answer this question, we need
additional knowledge about the geography of Bolivia and its neighboring countries.

Q: <question>
Knowledge Triplets:  <triplets>
```

Figure 18: Prompt used to generate the final answer, conditioned on the reasoning paths, in Think on Graph.

```
Q: What state is home to the university that is represented in sports by George
Washington Colonials men's basketball?
A: First, the education institution has a sports team named George Washington Colonials
men's basketball in is George Washington University , Second, George Washington
University is in Washington D.C. The answer is Washington, D.C..

Q: Who lists Pramatha Chaudhuri as an influence and wrote Jana Gana Mana?
A: First, Bharoto Bhagyo Bidhata wrote Jana Gana Mana.  Second, Bharoto Bhagyo Bidhata
lists Pramatha Chaudhuri as an influence.  The answer is Bharoto Bhagyo Bidhata.

Q: Who was the artist nominated for an award for You Drive Me Crazy?
A: First, the artist nominated for an award for You Drive Me Crazy is Britney Spears.
The answer is Jason Allen Alexander.

Q: What person born in Siegen influenced the work of Vincent Van Gogh?
A: First, Peter Paul Rubens, Claude Monet and etc.  influenced the work of Vincent Van
Gogh.  Second, Peter Paul Rubens born in Siegen.  The answer is Peter Paul Rubens.

Q: What is the country close to Russia where Mikheil Saakashvii holds a government
position?
A: First, China, Norway, Finland, Estonia and Georgia is close to Russia.  Second,
Mikheil Saakashvii holds a government position at Georgia.  The answer is Georgia.

Q: What drug did the actor who portrayed the character Urethane Wheels Guy overdosed
on?
A: First, Mitchell Lee Hedberg portrayed character Urethane Wheels Guy.  Second,
Mitchell Lee Hedberg overdose Heroin.  The answer is Heroin.

Q: <question>
```

Figure 19: Prompt used to generate the final answer, without reasoning paths, in Think on Graph.

```
You are a QA bot.
You will be given a question.
Respond to the question to the best of your knowledge.
Respond with the single most accurate answer.
Question may have multiple answers, but you must choose only one.
Respond with the answer you consider most correct.
Be as precise as you can.

Respond with only the answer, and no other words.
DO NOT respond with a full sentence.
DO NOT add notes.
DO NOT add any symbols before each answers.

Example Input Format:
Question:  What color is the sky?

Example Response:
Blue

Question:  <question>
```

Figure 20: Prompt used to generate the final answer in Plain Question.

D. Multiple Answers Experiments

This section reports the results of the experiments conducted with GCR and RoG methods, allowing the model to generate multiple answers per question.
Table 16 reports the average and median number of predictions generated per question.
Table 17 reports the Adherence, Resistance, and Incorrectness metrics computed as micro-averages (average among all predictions).
Table 18 reports the Adherence, Resistance, and Incorrectness metrics computed as macro-averages (average of the averages of each question).

Model	Method	Dataset	Total number of predictions	Total number of questions	Mean number of predictions	Median number of predictions
nano	PQ	Original	3564	1462	2,44	2,00
	GCR	Original	3811	1462	2,61	2,00
		Slight	3667	1462	2,51	2,00
		Significant	3567	1462	2,44	2,00
		Comical	3674	1462	2,51	2,00
		Uncomp	3611	1462	2,47	2,00
	RoG	Original	6038	1462	4,13	2,00
		Slight	6201	1462	4,24	2,00
		Significant	5959	1462	4,08	2,00
		Comical	6609	1462	4,52	2,00
		Uncomp	5444	1462	3,72	2,00
standard	PQ	Original	268	86	3,12	1,00
	GCR	Original	238	86	2,77	2,00
		Slight	261	86	3,03	3,00
		Significant	249	86	2,90	3,00
		Comical	226	86	2,63	2,00
		Uncomp	254	86	2,95	3,00
	RoG	Original	508	86	5,91	2,00
		Slight	557	86	6,48	2,00
		Significant	479	86	5,57	2,00
		Comical	436	86	5,07	2,00
		Uncomp	490	86	5,70	2,00

Table 16: In the GCR and RoG methods, the question is given to an LLM together with the generated paths; in PQ, the question is presented without additional information (so it is applied only on Original dataset). Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). In these experiments, all methods are allowed to generate multiple final answers. Table shows total number of predictions generated, and the mean and median number of predictions generated per question for both models (GPT-4.1-standard and GPT-4.1-nano), all methods (PQ, GCR and RoG) and all datasets.

Model	Method	Dataset	Total number of predictions	Adherence %	Resistance %	Incorrectness %
nano	PQ	Original	3564	49%	0%	51%
	GCR	Original	3811	76%	0%	24%
		Slight	3667	57%	7%	37%
		Significant	3567	50%	6%	44%
		Comical	3674	58%	6%	35%
		Uncomp	3611	29%	11%	60%
	RoG	Original	6038	68%	0%	32%
		Slight	6201	48%	6%	46%
		Significant	5959	43%	5%	52%
		Comical	6609	49%	6%	45%
		Uncomp	5444	21%	8%	71%
standard	PQ	Original	268	60%	0%	40%
	GCR	Original	238	72%	0%	28%
		Slight	261	55%	3%	42%
		Significant	249	49%	3%	48%
		Comical	226	58%	4%	39%
		Uncomp	254	35%	7%	58%
	RoG	Original	508	52%	0%	48%
		Slight	557	43%	2%	55%
		Significant	479	46%	3%	51%
		Comical	436	56%	3%	41%
		Uncomp	490	48%	3%	50%

Table 17: In the GCR and RoG methods, the question is given to an LLM together with the generated paths; in PQ, the question is presented without additional information (so it is applied only on Original dataset). Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). In these experiments, all methods are allowed to generate multiple final answers. Table shows Adherence, Resistance and Incorrectness metrics computed as micro-averages (i.r. metrics computed on all predictions, aggregating all questions) for both models (GPT-4.1-standard and GPT-4.1-nano), all methods (PQ, GCR and RoG) and all datasets.

Model	Method	Dataset	Total number of questions	Adherence %	Resistance %	Incorrectness %
nano	PQ	Original	1462	52%	0%	48%
	GCR	Original	1462	78%	0%	22%
		Slight	1462	58%	8%	34%
		Significant	1462	49%	8%	43%
		Comical	1462	59%	7%	34%
		Uncomp	1462	29%	14%	58%
	RoG	Original	1462	70%	0%	29%
		Slight	1462	51%	11%	37%
		Significant	1462	48%	7%	43%
		Comical	1462	50%	11%	38%
		Uncomp	1462	22%	13%	63%
standard	PQ	Original	86	68%	0%	32%
	GCR	Original	86	75%	0%	25%
		Slight	86	58%	3%	39%
		Significant	86	51%	3%	45%
		Comical	86	63%	4%	33%
		Uncomp	86	37%	9%	54%
	RoG	Original	86	68%	0%	32%
		Slight	86	55%	6%	39%
		Significant	86	56%	6%	38%
		Comical	86	59%	6%	34%
		Uncomp	86	54%	9%	37%

Table 18: In the GCR and RoG methods, the question is given to an LLM together with the generated paths; in PQ, the question is presented without additional information (so it is applied only on Original dataset). Answers are classified as adherent if they match the provided context, resistant if they ignore the context but match the ground truth, and incorrect otherwise (see Subsection 6.2 for details). In these experiments, all methods are allowed to generate multiple final answers. Table shows Adherence, Resistance and Incorrectness metrics computed as macro-averages (i.e. metrics computed per single question, then averaged across all questions) for both models (GPT-4.1-standard and GPT-4.1-nano), all methods (PQ, GCR and RoG) and all datasets.

Abstract in lingua italiana

I Large Language Model (LLM) combinano la conoscenza parametrica acquisita durante l'addestramento con la conoscenza contestuale fornita all'interno del prompt; tuttavia, spesso faticano a sovrascrivere preconcetti errati quando vengono presentate evidenze contraddittorie. Questa tesi indaga questo "tiro alla fune" nel contesto della Retrieval-Augmented Generation (RAG) basata su Knowledge Graph (KG). Utilizzando un dataset di question answering accompagnato da un Knowledge Graph, vengono costruite quattro varianti progressivamente modificate (Slight, Significant, Comical and Uncomprehensible) introducendo contraddizioni controllate all'interno del KG. Tre metodi di RAG basati su KG, Graph Constrained Reasoning (GCR), Reasoning on Graphs (RoG) e Think on Graph (ToG), vengono valutati in termini di qualità dei percorsi di ragionamento, accuratezza delle risposte e aderenza alla conoscenza contestuale, misurata attraverso Prior Bias e Context Bias. I risultati sperimentali mostrano che ToG è inaffidabile a causa di frequenti fallimenti nella generazione dei percorsi, mentre GCR supera costantemente RoG producendo solo percorsi validi e raggiungendo una maggiore aderenza alle informazioni contestuali. Sebbene la RAG basata su KG migliori significativamente la fattualità delle risposte rispetto a una baseline senza recupero di informazioni aggiuntive, l'aderenza al contesto contraddittorio rimane parziale e peggiora con l'aumentare delle perturbazioni. Questi risultati evidenziano sia i benefici sia i limiti del recupero di informazioni strutturate nel guidare il ragionamento degli LLM in presenza di conoscenza conflittuale.

Parole chiave: Large Language Models, Retrieval-Augmented Generation, Knowledge Graphs, mitigazione delle allucinazioni, conoscenza contestuale, conoscenza parametrica.

Acknowledgements

Ringrazio il Prof. Marco Brambilla e il Dr. Antonio De Santis per l'assistenza e la guida durante questo percorso.
Ringrazio i miei amici, presenti e passati, per tutti i momenti condivisi negli anni.
Ringrazio mio fratello Filippo, nonna Enrica, nonno Silvio, e tutti i miei familiari, per l'affetto e il sostegno costanti.
Un grazie speciale a mamma e papà: vi voglio bene.
Ed un ricordo allo zio Massimo.