Improving LLM Accuracy with Knowledge Graphs in Solving Algebra Problems

SHUN NISHIJIMA, University of Twente, The Netherlands

Large Language Models (LLMs) have demonstrated remarkable capabilities in many real-world applications. Recent models also show high performance in solving simple calculations. However, LLMs are often criticized for producing hallucinations when solving problems that require accurate symbolic reasoning, such as algebra tasks. Some researchers have explored approaches that use large knowledge bases or fine-tuning to solve mathematical problems. Others have explored using structured knowledge to generate one-shot answers to general knowledge questions. Still, there is little research on using small, flexible, structured knowledge to improve LLMs' mathematical reasoning. This research proposes a lightweight, domain-specific knowledge graph (KG)-based prompting method to enhance LLM accuracy in solving high school algebra problems. A manually constructed KG containing key algebraic concepts and procedures is injected into the prompts to provide a structured context. Using 40 algebra problems and two LLMs (GPT-4o and GPT-4.1-mini), the study demonstrates that KG-based prompts improve average factual accuracy from 61.5% to 73.75% with statistically significant gains, particularly in expression simplification tasks. These results suggest that small, targeted KGs may serve as an effective, low-cost alternative to improve reasoning accuracy in LLMs without requiring retraining or external tools.

Additional Key Words and Phrases: Large Language Models, Knowledge Graphs, Prompt Engineering, Algebra, Mathematical Reasoning

1 INTRODUCTION

Large Language Models (LLMs), such as GPT-4 [15], have shown significant performance in a wide range of real-world applications. Experiments demonstrate that GPT-4 is capable of performing a diverse set of tasks, including mathematics, coding, vision, and logical reasoning. GPT-4 has also been observed to show human-like performance in areas such as programming and problem solving [3]. LLMs are also being applied in sensitive domains such as healthcare. GPT-4 can answer complex medical questions, such as those of medical licensing exams. Performance is even better than that of medical students in many standardized tasks. In particular, on multiple-choice clinical questions, the model shows high precision [14]. In addition, its efficiency, scalability, transfer learning, and cross-domain utility can present potential for future applications [2]. LLMs have also demonstrated capabilities in mathematical problem-solving, particularly in simple algebra. Recent studies show that ChatGPT performs well on simple calculations. Despite their impressive fluency and reasoning abilities, LLMs are criticized for their limitations in handling factual accuracy and their tendency to generate hallucinations, especially when they handle complicated algebra problems, which require multi-step symbolic reasoning. While models can mimic mathematical reasoning, they often make logical or calculation errors without external tools [9].

Reducing these errors is important, particularly for educational applications that need high accuracy and logical consistency. ChatGPT's ability is behind that of trained educators in accuracy and consistency in identifying some common student errors. Because models frequently lack context-specific understanding, they are not completely reliable in the educational application [1].

Currently, there is limited literature on the method that improves the performance of LLMs specifically in processing accurate symbolic reasoning. In real-life applications, there are a few existing examples of lightweight, computationally efficient methods, although they remain underexplored. This research investigates how domain-specific knowledge graph (KG)-based prompts can help LLMs' reasoning. The primary goal of this research is to examine whether KGs incorporated prompts can significantly improve LLMs' factual accuracy in solving algebra problems compared to standard output.

To achieve the goal, the following research questions (RQs) will be addressed:

(1) RQ1: To what extent does incorporating domain-specific KG-based prompts improve the factual accuracy of LLM outputs in solving algebra problems compared to standard prompting?
(2) RQ2: How does the impact of KG-based prompting on factual accuracy differ between GPT-4.1-mini and GPT-4o in solving algebra problems?
(3) RQ3: What types of algebra problems benefit most from KG-augmented prompts by reducing hallucinations and logical errors?

The structure of this paper is as follows: Section 2 provides an overview of the related academic literature, shows LLMs' capabilities, existing solutions to extend their abilities in the mathematical area, and identifies gaps between existing solutions and demands. Section 3 represents the definition of keywords and the specified area of algebra used in this research to set the domain manipulated. Section 4 outlines the methodology that will be used to address the research questions, including the way of using KGs in prompts and how to evaluate their outputs. Section 5 explains the experiment setup and the results from these experiments by automation software and a statistical review of the results. Section 6 provides a discussion of the findings, their implications. The final section (Section 7) concludes the paper by summarizing the contributions, explaining limitations of the current approach, and providing directions for future work.

2 RELATED WORK

Numerous studies have examined the capabilities and limitations of LLMs, particularly their tendency to generate inaccurate or inconsistent outputs. To address these challenges, researchers have developed various strategies to enhance model reliability, including improved prompting methods for mathematical reasoning. Among

TScIT 43, July 4, 2025, Enschede, The Netherlands

these, techniques that incorporate structured knowledge, such as KG, have shown notable success in reducing reasoning errors. These prior efforts provide a foundation for this research, which builds on structured prompting to explore the integration of domain-specific KG for solving algebra problems.

2.1 Capabilities and Limitations of LLMs

LLMs such as GPT-3.5, GPT-4, and Google's PaLM 2 have shown strong abilities in various domains, including natural language understanding, programming, medical, and mathematical reasoning [3, 14]. GPT-4, the latest model of GPT, demonstrates stronger reasoning, coding, and language abilities than the previous models [15].

Particularly, recent research has highlighted their ability to perform symbolic tasks such as arithmetic, algebra, and calculus with varying degrees of accuracy. Despite these successes, their ability falls short on tasks requiring precise symbolic manipulation or strict logical consistency. Especially, in the GPT-4 case, their performance declines on complex tasks that require deep reasoning or creative problem-solving. ChatGPT also makes hallucinations, skips steps, and uses invalid logic. This performance gap is attributed to their lack of explicit domain knowledge and reliance on patterns learned from data rather than a deep understanding of mathematical principles [4, 9].

Although LLMs can mimic reasoning through prompt engineering, the output is not guaranteed to align with formal logic or mathematical rules [9]. These issues highlight the need for mechanisms that can incorporate structured, domain-relevant knowledge into the prompting process of LLMs.

2.2 Prompting Techniques for Mathematical Reasoning

Researchers have explored various strategies to moderate the tendency to hallucinate through techniques. Chain-of-thought (CoT) prompting guides the model in explicitly generating intermediate steps to get a final answer. This technique simulates how the human brain works by breaking down the main question into small tasks. It demonstrates large gains in performance for math by reducing hallucinations, but it still has limitations in producing correct intermediate steps since it works without external contextual knowledge [10, 19].

Self-consistency sampling is a method that improves CoT prompting by introducing multiple reasoning paths to solve the problem and selecting the most consistent answer. Results show significantly higher accuracy on tasks like grade-school math and logical reasoning. However, it requires a high computational cost by producing multiple paths for each query. The method also does not guarantee a logically correct answer because majority voting is selected as the answer [18].

Another research, verification-based models, are built by generating multiple candidate solutions and evaluating them based on correctness or logical consistency. Using a separate verifier model to assess the correctness of multiple candidate answers, the model shows a significant reduction in hallucination or incorrect responses. While this method is effective in comparison with fine-tuning, the strategy is computationally intensive and may not scale efficiently without optimization [4].

These strategies aim to improve intermediate reasoning and show success in enhancing mathematical problem-solving abilities. However, they still face limitations in contextual understanding, specific domain knowledge, and computational costs.

2.3 Knowledge Graphs and Structured Contexts

Providing structured knowledge, models can process mathematical reasoning, reducing hallucinations and logical errors. One promising direction involves using KGs to enable more structured reasoning. For example, KnowGPT shows that integrating structured domain knowledge from KGs can enhance their performance on fact-intensive tasks. The method that directly integrates domain-specific KGs into LLM prompting is lightweight and effective in improving reasoning consistency. However, the large graph can be computationally expensive [21].

Similarly, Think-on-Graph represents the idea that introducing external KGs in LLM reasoning can lead to more interpretable and logical outputs. While this method improves interpretability and traceability of logic, it requires well-structured, explainable logic, which is not ideal for a low-resource environment [23].

Additionally, MindMap explores graph-of-thought structures derived from KGs to structure LLM reasoning. While the method made progress in improving transparency and logical flow in complex problem-solving, it heavily relies on well-formed KGs that need careful preparations [20].

In the recent work, GraphRAG provides a hybrid strategy of KG-based and Retrieval-augmented generation (RAG) framework that supports an iterative method to retrieve knowledge from the graph. It can support deeper, multi-step reasoning. Therefore, it outperforms previous graph-integrated methods. However, the framework brings high complexity and slow inference [12].

Specifically for mathematics, Math-KG introduces a large-scale mathematical KG integration to improve learning outcomes. The method to construct pipelines with mathematical knowledge bases for artificial intelligence (AI) works well [22]. Likewise, fine-tuning LLMs with a specific dataset can enhance the reliability of single-shot answers [11].

However, fine-tuning and the large-scale KGs are resource-heavy and less flexible for domain-specific tasks like algebra. Other prompt engineering methods are also computationally expensive due to using a large KG. Therefore, there is a growing interest in lightweight KG-based prompting methods, which offer a scalable and cost-effective alternative to fine-tuning or model retraining. The most related benchmark to this research is KnowGPT, which directly injects KGs into prompts, but not specified in the mathematics, particularly algebra problems.

2.4 Research Gap

Regarding insights from the above sub-sections, several methods demonstrate benefits for improving LLM performance in the mathematical area. However, they also present several limitations.

First, while CoT, SC, and verification-based strategies improve reasoning to some extent, they lack explicit domain knowledge, often failing on problems requiring formal symbolic manipulation.

Second, large-scale KGs are not always feasible in practice, particularly for specific educational domains or deployment environments with limited resources. Few studies have examined the potential of small-scale, domain-specific KGs specified to high school-level algebra, which represents a practical and accessible application area.

In addition, the methods for incorporating KGs into prompts vary widely. For example, KnowGPT directly injects KG into the model input prompt; Think-on-Graph and Mind-map queries on graph-shaped knowledge; and GraphRAG integrates iterative KG-guided retrieval into an RAG framework. However, compared to direct injection of KG into prompt, graph-structured reasoning and RAG methods often require additional infrastructure, retraining, or retrieval components.

To address these gaps, this research proposes a lightweight alternative made of a small and algebra-specific KG-based prompting. It aims to enhance LLM performance without retraining and constructing a large knowledge base, which can achieve a scalable way to improve factual accuracy in algebra problem solving.

3 BACKGROUND

This section describes the fundamental concepts that define the scope of this research, including the algebraic domain, the role of KGs, and the application of prompt engineering.

3.1 Focus Areas in Algebra

This research focuses on fundamental algebra topics typically taught at the high school level, particularly those that require multi-step symbolic reasoning. Two representative problem types are selected: factoring trinomials and simplifying expressions. Factoring tasks involve decomposing quadratic expressions into binomial products, such as transforming $x^2 + 5x + 6$ into $(x + 2)(x + 3)$. Simplifying expressions refers to reducing polynomials or rational expressions to their simplest forms. These problem types are well-suited for this research because they require symbolic manipulation, a process where reasoning errors frequently occur when intermediate steps are skipped or misapplied.

3.2 Knowledge Graph

A knowledge graph (KG) represents information in a structured format using nodes (concepts) and edges (relations). For this research, a small-scale, domain-specific KG is constructed (see Appendix). The triple structure of KGs enables lightweight and flexible use for a variety of use cases [7].

3.3 Prompt Engineering

Prompt engineering refers to the practice of designing the inputs to LLMs that improve their responses. In this research, KG triples are injected into the prompt to provide the model with relevant algebraic concepts and reasoning steps [5].

4 METHODOLOGIES

This section outlines the experimental design used to evaluate whether KG-based prompts improve the factual accuracy of LLMs when solving algebra problems. The methodology includes five key components: KG construction, prompt design, problem selection, model configuration, and evaluation strategies. Two models, GPT-4o and GPT-4.1-mini, are used to solve problems under two prompt conditions: standard and KG-based. Model outputs are evaluated for factual correctness using symbolic computation tools, and the results are analyzed using a statistical method to show significance.

4.1 KG Construction

A small and specific knowledge graph is constructed to represent core algebra concepts relevant to factoring and simplifying expressions. The KG is structured as triples (subject-predicate-object) and derived from an educational platform, OpenStax [17].

One of the subdomains, factoring trinomials, has its focused subgraph consisting of approximately 30 nodes. The other subdomain, simplifying expressions, has an extended graph from factoring rules because the factoring rule is also used for simplifying expressions; therefore, the total size of nodes is 60, which includes the factoring subgraph. Concepts such as "Factoring Trinomial includes step Finding factor pair (p,q) such that $p + q = b$" are formalized into triples. Additionally, worked examples of algebraic transformations are included as nodes to guide LLM reasoning.

4.2 Prompt Design

Two types of prompts are used in this study: a standard prompt and a KG-based prompt. In both cases, a list of problems follows an instruction that explicitly asks the model to solve the problems through symbolic reasoning, without using computational tools. The standard prompt begins with the directive:

> *For problems, factor each of the following by manual calculations without using code or programming tools like SymPy.*

This phrasing is intended to discourage the model from using code-based solutions and instead encourage step-by-step factorization procedures, similar to how a student would approach such tasks manually. This setup ensures a fair evaluation of the model's symbolic reasoning abilities under each prompting condition. A list of algebra problems immediately follows this instruction. Similarly, the prompts for simplifying expressions begin with the instruction:

> *For the following exercises, simplify the expression by manual calculations without using code or programming tools like SymPy.*

This instruction ensures that the model performs explicit simplification steps, rather than relying on computational shortcuts. It enables a clearer evaluation of whether KG-based prompts enhance the model's reasoning ability in symbolic algebra.

In the KG-based prompt, each input begins with the phrase:

> *Based on the knowledge below,*

This phrase is followed by a list of structured triples derived from the knowledge graph. After presenting the KG, the same instructional

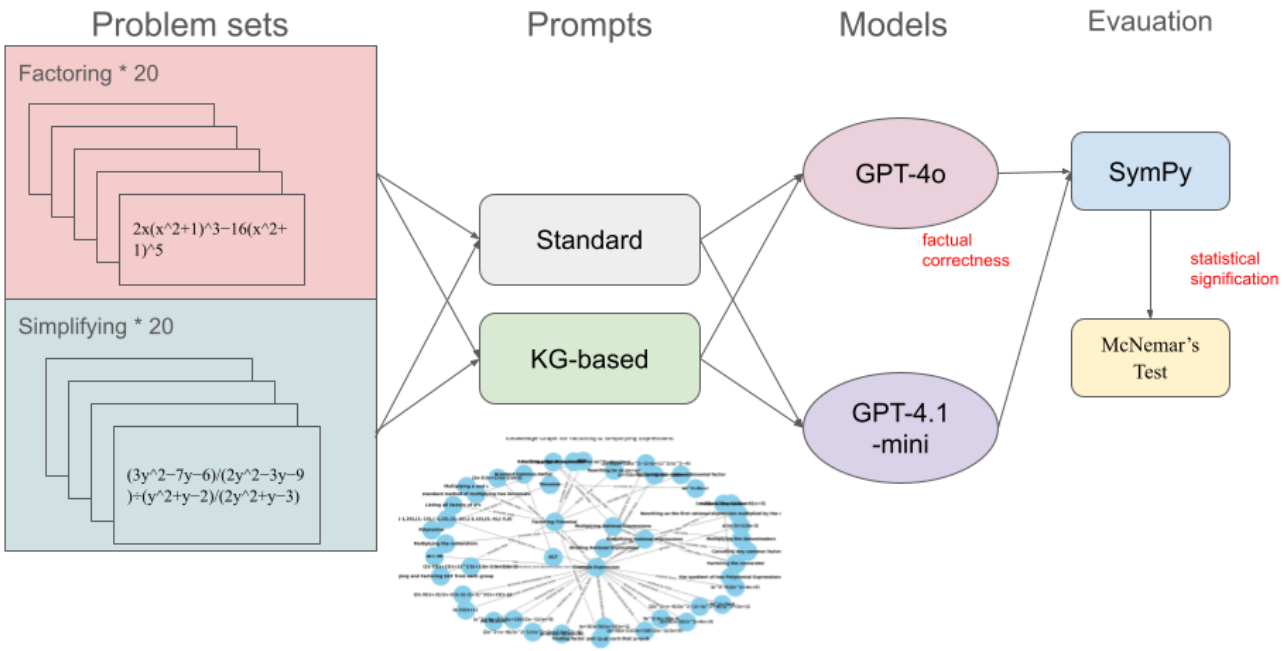

Fig. 1. Methodological Overview

phrase and corresponding problem set are provided as in the standard format. This method is applied consistently across both problem types, factoring and simplifying, by injecting domain-relevant subgraphs. The goal of this approach is to provide lightweight, symbolic reasoning support that mimics worked examples, without requiring model retraining or external retrieval mechanisms.

4.3 Problem Set Design

The algebra problems used in this study were sourced from two open educational platforms: OpenStax [17] and Paul's Online Math Notes [6]. OpenStax is a comprehensive, college-level textbook that covers foundational topics in algebra and trigonometry. It is commonly used in high school and undergraduate mathematics courses. Paul's Online Math Notes is a free online resource that offers tutorials in algebra. The final dataset includes 40 problems, 20 for factoring and 20 for simplifying rational expressions. All selected problems require multi-step reasoning, making them suited to evaluating whether giving knowledge can help LLMs provide correct answers. A detailed list of the problems used is in the Appendix.

4.4 Model Configuration

The experiment involves two language models: GPT-4o, which represents the most recent and advanced version in the GPT series, and GPT-4.1-mini, a smaller and cost-effective variant [16]. Both models are queried in instant chat mode with memory disabled to ensure that no information carries over between trials. Each model is asked to solve all 40 problems under both the standard and KG-based prompting. To ensure the stability and reliability of the results, the full experiment is repeated across five independent trials per model.

4.5 Evaluation Strategy

Each model's output is evaluated based on factual correctness. This evaluation is automated using `SymPy`, a Python-based symbolic mathematics library that parses and compares expressions. Unlike numerical libraries that approximate values, SymPy performs symbolic reasoning to determine whether two expressions are algebraically equivalent [13]. From the result of correctness, the correctness rate is computed for each prompt condition and model combination.

McNemar's test is applied to determine whether the observed differences between the standard and KG-based prompts are statistically significant [8]. This test is used to analyze paired binary data. This method is particularly suited for evaluating outcomes under two different conditions applied to the same set of items. In this research, it is used to assess whether the number of correct and incorrect responses differs significantly between the two prompt types for each language model.

The test specifically compares discordant outcomes, when a model answered a problem correctly under one prompt type but not the other, using a 2×2 contingency table. The test statistic is calculated as:

$$\chi^2 = \frac{(b-c)^2}{b+c}$$

where b is the number of instances where the model was correct with the standard prompt and incorrect with the KG-based prompt, and c is the number of instances where the model was incorrect with the standard prompt and correct with the KG-based prompt. A resulting p-value less than 0.05 is considered statistically significant.

5 EXPERIMENTS AND RESULTS

This section presents the results of experiments conducted to evaluate the impact of KG-based prompts on the performance of LLMs in solving algebra problems. The analysis compares the correctness rates of two models, GPT-4o and GPT-4.1-mini, under two prompts, standard and KG-based. Factual correctness is evaluated using automated tools and statistical testing to determine the significance of observed differences.

5.1 Experiment Procedure

The overall experimental workflow consists of four main components: KG Construction, Prompt Construction, Model Querying, and Result Evaluation. The KG Construction phase was completed manually by extracting algebraic rules and examples from educational resources such as OpenStax. These were constructed into triples that encoded conceptual and procedural knowledge in algebra. During the Prompt Construction phase, standard and KG-based prompts were manually created. While the KG-based prompt added a relevant subset of triples from the KG before the problem statement, the standard prompt included the algebra problem following a simple instruction. The Model Querying was also performed manually using the ChatGPT web interface. Each type of problem was entered into ChatGPT under both prompting conditions, using GPT-4o and GPT-4.1-mini. Memory function was disabled to prevent interference between sessions. The process was repeated over five trials to account for variability; therefore, 800 results were recorded (20 problems × 2 types × 2 prompts × 2 models × 5 trials). The Result Evaluation phase was automated using the Python library SymPy to check factual correctness. Answers were evaluated for mathematical equivalence to the correct answer. The evaluation results were aggregated to compute correctness rates. To test whether differences in performance between two prompts are statistically significant, McNemar's test was applied using the Python library (see Appendix).

5.2 Results

Table 1 summarizes the accuracy rates of models across different problem types under both prompting conditions. The improvement score between the prompts is calculated. Interpretation of significance from the p-value can also be seen in the table. The results show that injecting domain-specific KGs into prompts leads to higher factual accuracy in solving algebra problems.

The use of KG-based prompts resulted in a statistically significant improvement of 12.25% in the overall accuracy of GPT-4 (All) models, increasing from 61.50% to 73.75% ($p < 0.01$). GPT-4o showed the largest benefit with a 21.50% increase with statistical significance ($p < 0.01$), while GPT-4.1-mini, although it already has a high baseline accuracy, only showed a minor improvement of 3.00%, which is not statistically significant ($p = 0.327$).

In terms of problem categories, simplifying expressions gained the most improvement from KG-augmented prompts, with an improvement score of 21.00% ($p < 0.001$). In contrast, factoring problems saw only a 4.50% increase, which does not reach statistical significance ($p = 0.324$).

These findings support the hypothesis that lightweight, domain-specific KGs can meaningfully enhance LLM reasoning in mathematical problem-solving, particularly for problems requiring deeper conceptual processing.

6 DISCUSSION

This section reflects on the experimental results presented in Section 5 and discusses their interpretations in the context of the RQs introduced earlier. The goal of this section is to interpret the quantitative results, evaluate the effectiveness of KG-based prompting, and critically analyze its practical value. By examining how the results answer the RQs, this section provides insights into the broader significance of the research.

6.1 Main Findings

This study provides clear evidence that directly injecting domain-specific, lightweight KGs into prompts can significantly improve the factual accuracy of LLMs in algebra problem-solving. In all evaluated aspects, across models and problem types, KG-based prompting performed better than the standard prompt. Overall accuracy increased from 61.5% to 73.75%, a 12.25% improvement, when KG-integrated prompts were used. This improvement is also statistically confirmed by a statistical test. This supports the hypothesis behind RQ1 that structured domain-specific knowledge enhances LLM reasoning, particularly in problems that require careful multi-step manipulation.

For RQ2, results also revealed that the benefit of KG prompting varies between models. GPT-4o, which had a lower baseline performance, showed a large gain of 21.5%, suggesting that less capable models benefit more from external structured knowledge. In contrast, GPT-4.1-mini, which already performed well under the standard prompt, showed only a 3% improvement, and the difference was not statistically significant.

Interestingly, although GPT-4o is the more advanced model, it showed a lower baseline accuracy (42%) than GPT-4.1-mini (81%) in this task. One possible reason for GPT-4o's lower baseline may be its sensitivity to instruction phrasing or an overreliance on learned language patterns in the absence of contextual structure. Further controlled studies would be needed to understand model behavior in symbolic domains. However, GPT-4o demonstrated a much larger improvement when KG-based prompts were introduced, jumping to 63.5%. This suggests that GPT-4o may be more responsive to structured input or more sensitive to contextual information.

The effect was huge in simplifying expressions, where accuracy increased from 47.5% to 68.5%, a statistically significant 21% improvement, suggesting that KG injection in prompts is especially effective in tasks that involve multi-step symbolic reasoning or abstract transformations. On the other hand, in factoring problems,

Model	Problem Type	Standard	KG-based	Improvement	p-value	Significance
GPT-4 (All)	All	61.50%	73.75%	12.25%	2.81E-06	Significant
GPT-4o	All	42.00%	63.50%	21.50%	2.30E-06	Significant
GPT-4.1-mini	All	81.00%	84.00%	3.00%	0.3267996	Not Significant
GPT-4 (All)	Simplifying	47.50%	68.50%	21.00%	6.63E-07	Significant
GPT-4 (All)	Factoring	75.50%	79.00%	4.50%	0.3239398	Not Significant

Table 1. Comparison of Standard vs. KG-based performance across models and problem types

the gain was only 4.5% and not statistically significant, although the baseline of the accuracy was relatively high (75.5%). This contrast addresses RQ3, that KG prompts are more helpful for problems with conceptual complexity rather than those based on repetitive structures.

6.2 Critical Reflection

A critical reflection reveals several limitations underlying these results. Despite the gains, the absolute accuracy of the models, especially GPT-4o, remains relatively low for the tasks that need high accuracy or educational use. For example, 63.5% accuracy with KG-based prompts may produce too many errors in a real-world high school classroom. This indicates that while KGs help reduce hallucinations, they do not fully resolve the reasoning gaps in LLMs.

Moreover, although small-scale KG is lightweight and flexible, the method depends on manual construction that may not scale easily across broader math domains or more advanced topics. In this case, the KG was built from a few definitions, procedural steps for resolution, and example usage of these steps, but this approach may not generalize to other domains. The results demonstrate effectiveness under controlled conditions, but how well the approach generalizes remains uncertain.

Additionally, while CoT or verification-based models enforce step-by-step logic, the KG-incorporated method passively injects structured context. As such, it likely improves internal representations, but does not ensure interpretability or formally correct reasoning. Compared to CoT prompting, which improves model accuracy on benchmarks like GSM8K (from 17.7% to 57.1%, +39.4%), the KG-based method achieved a notable +21.5% improvement for GPT-4o in algebra tasks. Although the prior study for the CoT method uses different models and different mathematical sample questions, the KG incorporated method showed comparable results without requiring an intermediate step. A combination of both may yield further gains [19].

In sum, KG-based prompting provides a flexible, low-resource alternative for improving LLM accuracy on symbolic math problems. It is most effective for problems requiring multi-step reasoning and beneficial for models with weaker reasoning capabilities. However, it is not a standalone solution, and further hybrid methods are likely needed to reach reliable levels of performance.

7 CONCLUSION

This research introduced the use of lightweight, domain-specific KGs to improve the factual accuracy of LLMs in solving algebra problems. By injecting small-scale KGs into prompts, the research aimed to reduce hallucinations and symbolic reasoning errors without using computational tools or fine-tuning. Experiments showed that KG-based prompting significantly improved model accuracy, particularly simplifying expressions, where symbolic errors are most common. GPT-4o is the model that benefited most from KG-based prompts. It suggests that structured context may enhance reasoning in advanced models. In contrast, GPT-4.1-mini showed only a modest improvement, likely due to its already high baseline performance. This study's KG-based prompting achieved accuracy improvements comparable to well-known strategies like CoT prompting without requiring multi-step output. These findings support the concept that structured prompts using domain knowledge may help reduce logical errors and hallucinations in LLMs, especially for tasks that involve multi-step symbolic reasoning.

7.1 Limitations

Although the findings suggest the potential of the KG-based method, several limitations affect the generalizability and strength of the findings. First, the study used only 40 problems (20 problems per type). The size of the samples affects the statistical confidence. This scope is further limited to two specific types, such as factoring and simplifying. This narrow scope may not fully represent the robustness in mathematical education that includes diverse and broader problems.

Second, the manual construction of KG introduces a subjective bias and limits scalability. This manual curation is not practical for larger or complex domains. The way of injecting structured knowledge into prompts is not always possible when larger inputs or many more outputs are expected.

Third, the evaluation solely focused on final answer correctness, as verified by symbolic equivalence in SymPy. This approach does not assess logical completeness or interpretability. In addition, there is no human baseline for students to try to solve by themselves, allowing them to compare LLM's capability and human capability.

7.2 Future Work

Future research should expand problem sets, such as equations, inequations, and functions, which would be useful for wider research in the mathematical area. These domains may reveal additional benefits or limitations of the KG-based method and suggest the availability of this method to wider or higher levels of mathematical solving. Applying to the domain outside of the mathematical domain is also considerable; the ideal domain requires logical reasoning steps, such as solving legal problems.

To extend interpretability, the hybrid method with existing prompt engineering solutions may reveal extra performance in logical correctness. For example, using the KG injection method with the CoT method, step-by-step instructions can be observed. The verification method can be applied to achieve higher correctness by majority voting among candidates may help filter out logically inconsistent responses.

Moreover, automation techniques for constructing small, domain-specific KGs may improve the method's generalizability. By introducing concrete logical conditions to the resource of structured knowledge, automatic retrieval of domain-specific KGs from each subdomain may be possible. Additionally, integrating KG-based prompts with LLM APIs could scale the experimentation process and support more diverse and extensive problem sets.

RAG or graph-embedded system may be the right direction for reaching high accuracy by an iterative process to think on the graph or search in documents. In mathematical terms, concepts and procedural solution steps can be applied to multiple types of problems; therefore, searching in larger connected sub-graphs may support model accuracy more.

Finally, targeting the educational applications of the KG-based method, compared to human students' outputs, may prove the method's availability in real-world applications.

REFERENCES

[1] Alexander Bewersdorff, Kai Lahnstein, Tomasz Dziubaniuk, Martin Hentschel, and Anna Busse. 2023. Can ChatGPT Reliably Identify Student Errors? A Comparison Between Large Language Models and Human Raters in Experimentation Protocols. *ArXiv* (2023). https://arxiv.org/abs/2308.06088

[2] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Sanjeev Arora, Sydney von Arx, and Percy Liang. 2021. On the Opportunities and Risks of Foundation Models. *ArXiv* (2021). https://arxiv.org/abs/2108.07258

[3] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, and Yoram Zhang. 2023. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. *ArXiv* (2023). https://arxiv.org/abs/2303.12712

[4] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Łukasz Kaiser, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *ArXiv* (2021). https://arxiv.org/pdf/2110.14168

[5] DAIR.AI. n.d.. Prompt Engineering Guide. https://www.promptingguide.ai/. Retrieved June 7, 2025.

[6] P. Dawkins. n.d.. Algebra - Paul's Online Math Notes. Lamar University. https://tutorial.math.lamar.edu/Classes/Alg/Alg.aspx.

[7] Lisa Ehrlinger and Wolfgang Wöß. 2016. Towards a Definition of Knowledge Graphs. In *SEMANTiCS*, Vol. 48. https://arxiv.org/abs/2003.02320

[8] Morten W. Fagerland, Stian Lydersen, and Petter Laake. 2013. The McNemar Test for Binary Matched-Pairs Data: Mid-p and Asymptotic Are Better Than Exact Conditional. *BMC Medical Research Methodology* 13, 91 (2013). https://doi.org/10.1186/1471-2288-13-91

[9] Simon Frieder, Lior Pinchuk, Ankit Kumar, Jordan Cotler, Romal Thoppilan, Deep Ganguli, and Barret Zoph. 2023. Mathematical Capabilities of ChatGPT. *ArXiv* (2023). https://arxiv.org/pdf/2301.13867

[10] Shuyuan Liu, Chunting Zheng, Yejin Bang, Hongyin Zhang, Xiang Ren, and Byron C. Wallace. 2023. What Makes Good In-Context Examples for GPT-3? *ArXiv* (2023). https://arxiv.org/abs/2302.11382

[11] Yujia Liu, Aman Singh, Charles D. Freeman, John D. Co-Reyes, and Peter J. Liu. 2023. Improving Large Language Model Fine-Tuning for Solving Math Problems. *ArXiv* (2023). https://arxiv.org/abs/2310.10047

[12] Shoukang Ma, Chen Xu, Xiang Jiang, Ming Li, Hongzhi Qu, Cheng Yang, Jiayuan Mao, and Jiafeng Guo. 2024. Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-Guided Retrieval Augmented Generation. *arXiv preprint arXiv:2407.10805* (2024). https://arxiv.org/abs/2407.10805

[13] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, and Brian E. Granger. 2017. SymPy: Symbolic Computing in Python. *PeerJ Computer Science* (2017). https://doi.org/10.7717/peerj-cs.103

[14] Harsha Nori, Nathan King, Scott M. McKinney, David Carignan, and Eric Horvitz. 2023. Capabilities of GPT-4 on Medical Challenge Problems. *ArXiv* (2023). https://arxiv.org/abs/2303.13375

[15] OpenAI. 2023. GPT-4 Technical Report. *ArXiv* abs/2303.08774 (2023). https://api.semanticscholar.org/CorpusID:257532815

[16] OpenAI. 2025. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/.

[17] OpenStax. n.d.. *Algebra and Trigonometry*. Rice University. https://openstax.org/details/books/algebra-and-trigonometry-2e

[18] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *ArXiv* (2022). https://arxiv.org/abs/2203.11171

[19] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, and Quoc V. Le. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *ArXiv* (2022). https://doi.org/10.48550/arXiv.2201.11903

[20] Yuyang Wen, Ziyang Wang, and Jianshu Sun. 2023. MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models. *ArXiv* (2023). https://arxiv.org/abs/2308.09729

[21] Qi Zhang, Cheng Chen, Yuan Liu, Wenhao Wei, Bo Wang, Chenguang Zhu, Xuanjing Huang, and Zhiyuan Liu. 2023. KnowGPT: Knowledge Graph Based Prompting for Large Language Models. *ArXiv* (2023). https://arxiv.org/abs/2312.06185

[22] Yusheng Zhang, Yuan Wang, Cheng Zhang, and Jiawei Zhang. 2022. Math-KG: Construction and Applications of Mathematical Knowledge Graph. *ArXiv* (2022). https://arxiv.org/pdf/2205.03772

[23] Yuxuan Zhou, Tian Xie, Kun Zhao, Shuo Yang, and Jian Wu. 2023. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. *ArXiv* (2023). https://arxiv.org/pdf/2307.07697

A APPENDIX

A.1 Knowledge Graphs

A.1.1 KG for Factoring Polynomials.

A.1.2 KG for Simplifying Expressions.

A.2 Prompts

A.2.1 Standard Prompt (Factoring). For problems, factor each of the following by manual calculations without using code or programming tools like SymPy.

- $y^2 + 16y + 60$
- $6t^2 - 19t - 7$
- $12t^2 + t - 13$
- $16x^2 - 100$
- $12x^2 + 31x + 7$
- $6z^2 - 35z + 36$
- $6u^8 - 3u^6 - 3u^4$
- $x^2 + 1 - 6x^{-2}$
- $x^4 - \frac{49}{x^2}$
- $9d^2 - 73d + 8$
- $90v^2 - 181v + 90$
- $2x(x^2 + 1)^3 - 16(x^2 + 1)^5$
- $18x - 2x^3 + 9 - x^2$
- $21 - w - 2w^2$
- $4x^6 + x^3 - 5$
- $2b^2 - 25b - 247$
- $w^2(1 + w^2)(8w - 1)^{10} + 9w(1 + w^2)^4(8w - 1)^7$
- $t^4 + 15t^3 + 14t^2$
- $5x(3x + 2)^{-2/4} + (12x + 8)^{3/2}$
- $z^2(4z - z^3) + 7(z^3 - 4z)$

A.2.2 KG-Integrated Prompt (Simplifying). Based on the knowledge below,

- Simplifying Rational Expressions - includes step - Factoring numerator

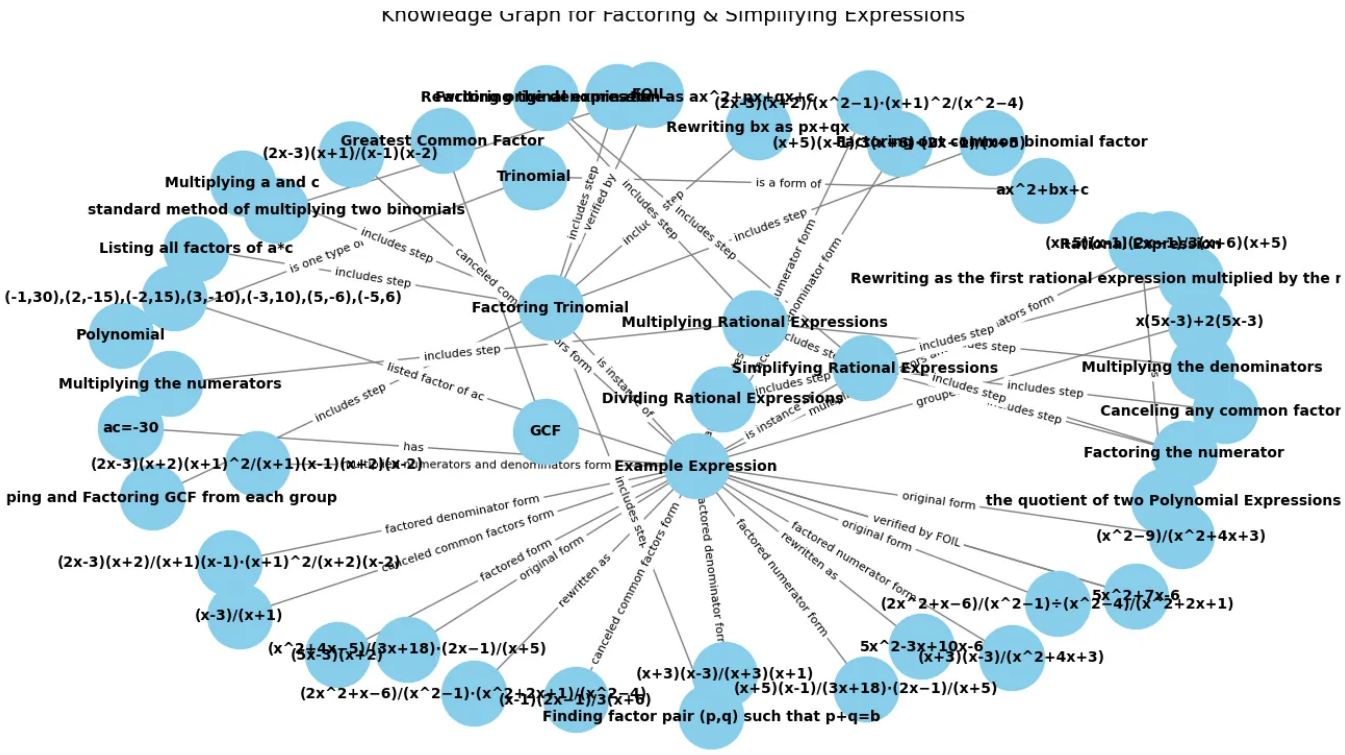

Fig. 2. KG visualized

Subject	Relation	Object
Trinomial	is a form of	$ax^2 + bx + c$
GCF	is	Greatest Common Factor
FOIL	is	Standard method of multiplying two binomials
Factoring Trinomial	includes step	Multiplying a and c
Factoring Trinomial	includes step	Listing all factors of $a \cdot c$
Factoring Trinomial	includes step	Finding factor pair (p, q) such that $p + q = b$
Factoring Trinomial	includes step	Rewriting bx as $px + qx$
Factoring Trinomial	includes step	Rewriting original expression as $ax^2 + px + qx + c$
Factoring Trinomial	includes step	Grouping and factoring GCF from each group
Factoring Trinomial	includes step	Factoring out common binomial factor
Factoring Trinomial	verified by	FOIL

Table 2. KG Triples for Factoring Trinomials

- Simplifying Rational Expressions - includes step - Factoring denominator
- Simplifying Rational Expressions - includes step - Canceling common factors

For the following exercises, simplify the expression by manual calculations without using code or programming tools like SymPy.

- $\frac{2a^2 - a - 3}{2a^2 - 6a - 8} \cdot \frac{5a^2 - 19a - 4}{10a^2 - 13a - 3}$
- $\frac{m^2 + 5m + 6}{2m^2 - 5m - 3} \div \frac{2m^2 + 3m - 9}{4m^2 - 4m - 3}$
- $\frac{4d^2 - 7d - 2}{6d^2 - 17d + 10} \div \frac{8d^2 + 6d + 1}{6d^2 + 7d - 10}$
- $\frac{6x^2 + 5x - 4}{3x^2 + 19x + 20}$

Subject	Relation	Object
Trinomial	is one type of	Polynomial
Rational Expression	is	The quotient of two Polynomial Expressions
Simplifying Rational Expressions	includes step	Factoring the numerator
Simplifying Rational Expressions	includes step	Factoring the denominator
Simplifying Rational Expressions	includes step	Canceling any common factors
Multiplying Rational Expressions	includes step	Factoring the numerator
Multiplying Rational Expressions	includes step	Factoring the denominator
Multiplying Rational Expressions	includes step	Multiplying the numerators
Multiplying Rational Expressions	includes step	Multiplying the denominators
Multiplying Rational Expressions	includes step	Simplifying Rational Expressions
Dividing Rational Expressions	includes step	Rewriting as multiplication with the reciprocal
Dividing Rational Expressions	includes step	Multiplying Rational Expressions
Dividing Rational Expressions	includes step	Simplifying Rational Expressions

Table 3. KG Triples for Simplifying Expressions

- $\frac{3c^2+25c-18}{3c^2-23c+14}$
- $\frac{2d^2+9d-35}{d^2+10d+21} \cdot \frac{3d^2+2d-21}{3d^2+14d-49}$
- $\frac{10h^2-9h-9}{2h^2-19h+24} \cdot \frac{h^2-16h+64}{5h^2-37h-24}$
- $\frac{6b^2+13b+6}{4b^2-9} \cdot \frac{6b^2+31b-30}{18b^2-3b-10}$
- $\frac{6x^2-5x-50}{15x^2-44x-20} \cdot \frac{20x^2-7x-6}{2x^2+9x+10}$
- $\frac{2n^2-n-15}{6n^2+13n-5} \cdot \frac{12n^2-13n+3}{4n^2-15n+9}$
- $\frac{36x^2-25}{6x^2+65x+50} \cdot \frac{3x^2+32x+20}{18x^2+27x+10}$
- $\frac{3y^2-7y-6}{2y^2-3y-9} \div \frac{y^2+y-2}{2y^2+y-3}$
- $\frac{q^2-9}{q^2+6q+9} \div \frac{q^2-2q-3}{q^2+2q-3}$
- $\frac{18d^2+77d-18}{27d^2-15d+2} \div \frac{3d^2+29d-44}{9d^2-15d+4}$
- $\frac{16x^2+18x-55}{32x^2-36x-11} \div \frac{2x^2+17x+30}{4x^2+25x+6}$
- $\frac{144b^2-25}{72b^2-6b-10} \div \frac{18b^2-21b+5}{36b^2-18b-10}$
- $\frac{16a^2-24a+9}{4a^2+17a-15} \div \frac{16a^2-9}{4a^2+11a+6}$
- $\frac{9x^2+3x-20}{3x^2-7x+4} \div \frac{6x^2+4x-10}{x^2-2x+1}$
- $\frac{x^2+x-6}{x^2-2x-3} \cdot \frac{2x^2-3x-9}{x^2-x-2} \div \frac{10x^2+27x+18}{x^2+2x+1}$
- $\frac{x^2+7x+12}{x^2+x-6} \div \frac{3x^2+19x+28}{8x^2-4x-24} \div \frac{2x^2+x-3}{3x^2+4x-7}$

B MCNEMAR'S TEST CODE

C MODEL ACCURACY RESULTS

D AI NOTICE

During the preparation of this work, the author used Grammarly for spelling and grammar correction and ChatGPT for fact-checking simplified explanations. All content was reviewed and verified by the author, who takes full responsibility for its accuracy.

```
from statsmodels.stats.contingency_tables import mcnemar

table = [[246, 295], [154, 105]]
result = mcnemar(table, exact=False)
print(f'Statistic: {result.statistic}, p-value: {result.pvalue}')
```

Fig. 3. Code used for McNemar's statistical test

Model	Prompt	Correct	Incorrect	Accuracy
GPT-4o	Standard	84	116	42.00%
GPT-4o	KG-based	127	73	63.50%
GPT-4.1-mini	Standard	162	38	81.00%
GPT-4.1-mini	KG-based	168	32	84.00%

Table 4. Accuracy Comparison Between Prompt Types and Models