CausalBench: A Comprehensive Benchmark for Evaluating Causal Reasoning Capabilities of Large Language Models

Zeyu Wang
University of California, Los Angeles
zeyuwang@ucla.edu

Abstract

Causal reasoning, a core aspect of human cognition, is essential for advancing large language models (LLMs) towards artificial general intelligence (AGI) and reducing their propensity for generating hallucinations. However, existing datasets for evaluating causal reasoning in LLMs are limited by narrow domain coverage and a focus on cause-to-effect reasoning through textual problems, which does not comprehensively assess whether LLMs truly grasp causal relationships or merely guess correct answers. To address these shortcomings, we introduce a novel benchmark that spans textual, mathematical, and coding problem domains. Each problem is crafted to probe causal understanding from four perspectives: cause-to-effect, effect-to-cause, cause-to-effect with intervention, and effect-to-cause with intervention. This multi-dimensional evaluation method ensures that LLMs must exhibit a genuine understanding of causal structures by correctly answering questions across all four dimensions, mitigating the possibility of correct responses by chance. Furthermore, our benchmark explores the relationship between an LLM's causal reasoning performance and its tendency to produce hallucinations. We present evaluations of state-of-the-art LLMs using our benchmark, providing valuable insights into their current causal reasoning capabilities across diverse domains. The dataset is publicly available for download at https://huggingface.co/datasets/CCLV/CausalBench.

1 Introduction

Causal reasoning, the ability to understand and infer causal relationships between variables, is a fundamental aspect of human cognition and plays a crucial role in decision-making, problem-solving, and learning (Pearl, 2009). For large language models (LLMs), causal reasoning refers to the ability to accurately identify, represent, and reason about causal relationships described in text, mathematical equations, or code snippets (Pearl, 2009). Developing strong causal reasoning abilities in LLMs is essential for progress toward artificial general intelligence (AGI), as it enables models to understand not just correlations but the underlying mechanisms driving outcomes (Fridman and Pearl, 2022). This understanding is crucial for making accurate predictions, generating insightful explanations, and adapting to new situations, as core components of AGI.

However, existing causal reasoning benchmarks have several limitations that hinder their ability to comprehensively evaluate the causal reasoning capabilities of LLMs. First, current benchmarks often focus on a single perspective of causal reasoning, such as cause-to-effect, lacking a multifaceted assessment that considers effect-to-cause reasoning and the impact of interventions. This narrow focus allows models to correctly answer causal questions by chance without truly understanding the underlying causal relationships (Kaushik et al., 2020). Second, current benchmarks are primarily text-based, lacking diversity in problem types, such as mathematical and coding problems that can encapsulate causal dependencies. Incorporating these diverse problem formats would enable a more robust evaluation of LLMs' capacity to reason about causality across various modalities. Third, the limited scale of existing benchmarks may not provide a sufficiently comprehensive assessment of LLMs' causal reasoning abilities due to the limited scale of the benchmark dataset.

To address these limitations, we propose CausalBench, a comprehensive benchmark for evaluating the causal reasoning capabilities of LLMs. CausalBench comprises four perspectives of causal reasoning for each scenario: cause-to-effect, effect-to-cause, cause-to-effect with intervention, and effect-to-cause with intervention. This multi-perspective approach mitigates the potential for correct answers

Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10), pages 143–151

by chance and provides a more accurate evaluation of LLMs' understanding of causal relationships. Moreover, CausalBench includes a diverse set of problem types spanning textual, mathematical, and coding domains, enabling a comprehensive assessment of causal reasoning abilities across different modalities. The benchmark consists of more than 60,000 problems and employs six evaluation metrics to measure LLMs' causal reasoning performance.

The major contributions of CausalBench are three-fold: (1) evaluating four causal reasoning perspectives per scenario to robustly assess causal understanding, (2) incorporating a diverse problem set spanning math, code, and natural language for cross-modal evaluation, and (3) implementing strict quality control measures, including a causal inference engine check and human expert review, to ensure the benchmark's validity and reliability. By addressing the limitations of existing benchmarks, CausalBench aims to provide a more comprehensive and accurate evaluation of the causal reasoning capabilities of LLMs, facilitating progress towards AGI.

2 Dataset Construction Process and Method

The construction of CausalBench involves three key steps: manual generation of initial test cases, scaling up using LLM such as GPT-4 Turbo, and quality control through causal inference engines together with human verification. Initially, we manually create a set of test cases covering four aspects of causal inference: (a) cause to effect, (b) effect to cause, (c) cause to effect with intervention, and (d) effect to cause with intervention to ensure a comprehensive evaluation of causal reasoning capabilities from different perspective. To expand the dataset, we then use GPT-4 Turbo with few-shot prompting, leveraging the model's ability to generate additional test cases that adhere to the desired format and cover the four causal inference aspects. The few-shot prompts are designed to guide GPT-4 Turbo in producing a diverse and extensive set of problems that maintain consistency with the manually generated cases. Afterward, we implement a quality control process involving validation through causal inference engines and review by human experts. The causal inference engines verify the logical consistency and correctness of the generated test cases, while human experts review and refine the dataset to maintain high standards of quality and relevance.

2.1 Workflow Overview

2.2 Manual Analysis and Generation

For the text problems of our Benchmark, we randomly selected 100 questions from the CLADDER dataset (Choshen et al., 2022) and manually analyzed them to determine their category within (1) inference from cause to effect, (2) effect to cause, (3) cause to effect with intervention, or (4) effect to cause with intervention. These perspectives represent different dimensions of causal reasoning: (1) Cause to the effect: Given the cause, what is the likelihood of the effect? (2) Effect to cause: Given the effect, what is the likelihood of the cause? (3) Cause to effect with intervention: If an intervention is added to the causal relationship, given the cause, what is the likelihood of the effect? and (4) Effect to cause with intervention: If an intervention is added to the causal relationship, given the effect, what is the likelihood of the cause?

After categorizing the selected cases from the CLADDER dataset, we expanded them by creating additional questions for the other three perspectives. For example, if a case was classified as "cause to effect", we generated corresponding questions for "effect to cause", "cause to effect with intervention", and "effect to cause with intervention" manually.

To correctly expand other perspective questions and their ground truths, we visualized the relationships between variables using causal diagrams and analyzed these relationships by calculating conditional probabilities. Causal diagrams represent variables as nodes and causal relationships as directed edges. For example, consider the following hypothetical scenario:

Imagine a self-contained, hypothetical world with only the following conditions, and without any unmentioned factors or causal relationships: Parents' intelligence has a direct positive effect on parents' social status and child's intelligence. Other unobserved factors has a positive direct effect on parents' social status and child's intelligence. If a child is intelligent, would it be more likely that this child had intelligent parents?

In this scenario, the causal diagram would have four nodes: Parents' intelligence, Parents' social status, Child's intelligence, and Other unobserved factors. There would be directed edges from Parents' intelligence to Parents' social status and

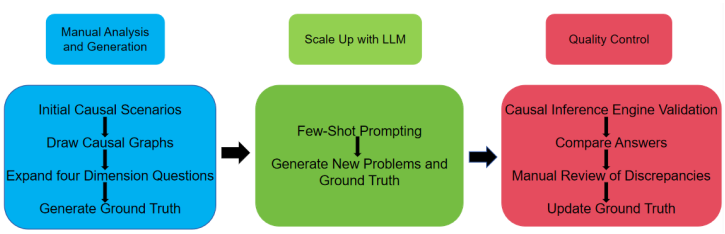

Figure 1: Workflow overview of the CausalBench dataset construction process.

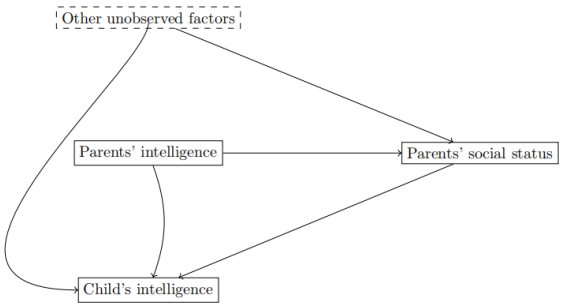

Figure 2: Causal Graph Example

Child's intelligence, from Other unobserved factors to Parents' social status and Child's intelligence, and from Parents' social status to Child's intelligence. Conditional probabilities can be estimated based on the causal graph.

Using the causal graph and conditional probabilities, we can categorized the original questions as effect-to-cause. The probability of the child being intelligent given that the parents are intelligent is higher than the probability of the child being intelligent given that the parents are unintelligent, so the ground truth is yes. Then extend the questions to cover four perspectives by adjusting the questioning logic and incorporating interventions into the causal path diagram, and calculate ground truth for each questions.(examples are provided in the Appendix)

Finally, we obtained 100 causal scenarios, with 400 causal questions. They serve as the foundation for our few-shot prompting approach, providing examples for GPT-4 Turbo on how to identify the type of the initial question and generate additional questions for the remaining perspectives. By using these examples in a few-shot prompting setting, we guide the model to generate additional perspective questions with answers for all other causal scenarios in the CLADDER dataset.

For coding and mathematical problems, we manually created 100 code scenarios and 100 math scenarios, each containing causal relationships, and designed four perspective questions for each scenario. These questions addressed causal issues based on the relationships described in the scenarios (examples are provided in the Appendix). We then used causal graphs and conditional probabilities to manually generate the ground truths and employed few-shot prompts with GPT-4 Turbo to generate additional code, math scenarios and questions with corresponding answers.

In summary, the manual analysis and generation process involved visualizing causal relationships using causal diagrams and calculating conditional probabilities for each scenario. We modified the questioning approach and added interventions to expand each problem into four forms, covering cause-to-effect, effect-to-cause, cause-to-effect with intervention, and effect-to-cause with intervention, and generated ground truths for each question. By the end of this section, we had created 100 sets of 400 text-based questions with ground truths, 100 sets of 400 coding questions with ground truths, and 100 sets of 400 math questions with ground truths. These manually generated samples serve as the foundation for our few-shot prompting approach, which utilizes GPT-4 Turbo to generate additional test cases.

2.3 Scaling Up with LLMs

After manually generating and verifying an initial set of questions, we employed GPT-4 Turbo to scale up the dataset. The scale-up process was divided into three parts: text problems, coding problems, and mathematical problems.

For the text problems, we provided GPT-4 Turbo with original CLADDER dataset(Choshen et al., 2022) questions with manually expanded questions along with their ground truths. By learning from these samples, GPT-4 Turbo was tasked with reading the remaining CLADDER scenarios (around 10,000 problems) and their corresponding questions, determining the question perspective, expanding the scenario into the other three perspec-

tives, and generating the associated ground truths. This process ensures every text causal scenario has four dimension questions and corresponding ground truths.

In the case of coding problems, we supplied GPT-4 Turbo with the 100 manually created code examples containing causal relationships. Using these examples as a foundation, GPT-4 Turbo generated an additional 2,000 code snippets, each incorporating causal relationships. For each newly generated code snippet, GPT-4 Turbo created four perspectives of questions and provided the corresponding ground truths, ensuring a comprehensive evaluation of causal reasoning in the context of programming.

Similarly, for mathematical problems, GPT-4 Turbo was employed to generate 2,000 new mathematical scenarios across various domains, such as probability theory, mathematical statistics, differential equations, and complex analysis. For each mathematical scenario, GPT-4 Turbo generated four types of questions and their associated ground truths, assessing the model's ability to reason about causal relationships in mathematical contexts.

By leveraging the capabilities of GPT-4 Turbo, we were able to create a dataset across all three problem categories. The text problems were augmented by automatically generating additional question perspectives and ground truths based on the existing CLADDER scenarios. The coding and mathematical problems were scaled up by having GPT-4 Turbo create new scenarios containing causal relationships and generate the corresponding questions and ground truths. This scale-up process resulted in a more comprehensive and diverse dataset, enabling a thorough evaluation of causal reasoning abilities in large language models across various domains.

2.4 Quality Control

2.4.1 Causal Inference Engine Design

To ensure the accuracy and consistency of the generated questions and answers, we developed a causal inference engine. This engine utilizes causal diagrams and conditional probabilities associated with each question to compute the answers for all questions. The causal inference engine serves as a verification layer, comparing the answers generated by the language model. If the answer generated by the language model differs from the answer generated by the causal inference engine, the case will be manually inspected, and the ground truth will be generated by human experts. Here are the Causal Inference Engine design details:

Input

- A causal scenario described in natural language, code, or mathematical equations, including causal relationships among variables, known conditions, etc.
- A causal query, which is a question based on causal scenario

Steps

Causal Graph Extraction:

For natural language scenarios, we identify variables and causal relationships, and construct causal graphs (G := (V, E)) by implementing a pipeline consisting of semantic parsing and coreference resolution modules. The semantic parsing module first uses the Stanford Parser (Klein and Manning, 2003) to perform syntactic parsing and obtain the sentence structure. Then, it applies Compositional Semantics (Zettlemoyer and Collins, 2005) to recursively map the syntactic parse tree to a logical form, based on the principle of compositionality. The coreference resolution module uses techniques such as the mention-pair model (Soon et al., 2001) to determine which mentions refer to the same entity, and merges the variables corresponding to coreferent mentions. From the outputs of the semantic parsing and coreference resolution modules, the pipeline automatically extracts variables from nouns and noun phrases, and identifies causal relationships indicated by verbs and conjunctions expressing causality (Li and Mao, 2019). Finally, the causal graph construction module takes the extracted variables as nodes (V) and causal relationships as directed edges (E) to automatically build the causal graph (Pearl, 2009).

For code scenarios, we identify variables and their dependencies, and construct causal graphs by implementing a pipeline that analyzes the code structure, control flow, and data flow. The pipeline first uses a code parser, such as the ast module (Python Software Foundation, 2023) in Python, to generate an abstract syntax tree (AST). It then performs control flow analysis using techniques like control flow graphs (CFGs) (Allen, 1970) and program dependence graphs (PDGs) (Ferrante et al., 1987), and data flow analysis using def-use chains (Harrold and Rothermel, 1994) and static single assignment (SSA) form (Cytron et al., 1991), to

identify execution paths, dependencies between statements, and variable dependencies. These analyses help automatically extract variables and their relationships from the code structure. Finally, the causal graph construction module takes the extracted variables as nodes (V) and their dependencies as edges (E) to build the causal graph based on the code semantics (Pearl, 2009), capturing the causal relationships between variables and enabling further reasoning and analysis.

For math scenarios, we identify variables and their functional relationships, and construct causal graphs by implementing a pipeline that parses and analyzes the mathematical equations. The pipeline first uses a math expression parser, such as the SymPy library (Meurer et al., 2017) in Python, to convert the equations into an abstract syntax tree (AST) representation. It then traverses the AST to identify variables and their functional relationships, such as dependencies and algebraic operations, using techniques like symbolic differentiation (Griewank and Walther, 2008) and expression simplification (Moses, 1971). These analyses help automatically extract variables and their relationships from the equation structure. Finally, the causal graph construction module takes the extracted variables as nodes (V) and their functional relationships as directed edges (E) to build the causal graph based on the equation semantics, similar to the approach in (Pearl, 2009). The resulting causal graph captures the causal relationships between variables in the mathematical equations, enabling further reasoning and analysis.

Query Classification: Classify the causal query into one of the three levels of the Ladder of Causation (Association, Intervention, Counterfactuals). Formalize the query into the corresponding causal language, as discussed in (Jin et al., 2023).

Estimand Derivation:

1. For text and math scenarios, we construct a module that uses causal inference algorithms (e.g., do-calculus (Pearl, 1995), counterfactual inference formulas (Pearl et al., 2000)) to derive the estimand based on the causal graph and query type.
2. For code scenarios, we use program analysis techniques (e.g., symbolic execution, data dependency analysis, control flow analysis) to derive the estimand based on the code structure and query type. This involve simulating interventions on code variables and analyzing the resulting program behavior.

Data Matching: Match the terms in the estimand with the available data or constraints in the scenario to obtain a computable estimand expression. Check the completeness and consistency of the data. Raise warnings or errors if critical data is missing. For code scenarios, this involve executing the code with specific inputs and observing the outputs. This step is similar to the data matching phase in (Jin et al., 2023).

Causal Effect Estimation:

1. Calculate the causal effect value based on the estimand expression and the available data, yielding the answer to the query.
2. For scenarios with unobserved confounders, use instrumental variable estimation (Angrist et al., 1996) or front-door adjustment (Pearl, 1995).
3. For code scenarios, this involve comparing program behaviors under different interventions.

This step is inspired by causal effect estimation phase in (Jin et al., 2023).

Output

- Answer to the causal query, including the estimated causal effect, confidence interval, and key assumptions.

In a summary, our Causal Inference Engine extends the original design presented in (Jin et al., 2023) by incorporating domain-specific graph extraction and estimand derivation techniques to handle causal inference problems in text, code, and math scenarios. The overall pipeline remains consistent with the one described in (Jin et al., 2023), but the internal methods are adapted to the specific structures and semantics of each domain.

2.4.2 Quality Control Process

After expansion with GPT4-Turbo, we obtained around 10000 x 4 text-based questions, 2000 x 4 math questions, and 2000 x 4 coding questions, along with their GPT-4 Turbo generated answers. To ensure the accuracy of the ground truth of each questions, we employed a strict quality control process as showing below:

We used the causal inference engine introduced above to independently solve the problems and generate its own set of answers. We compared the answers generated by GPT-4 Turbo and the causal inference engine. If two answers were the same,

we updated the answer as ground truth. If any of the answers were inconsistent, we conducted a manual analysis of the question and answers to determine the correct answer and update ground truth accordingly.

This multi-step quality control process, involving the use of causal inference engine and human expert check, ensures that the final dataset contains accurate and reliable questions and answers. The manual review of inconsistent answers further enhances the quality of the dataset by addressing any discrepancies or edge cases that the models may encounter.

3 Benchmark Results

3.1 Baseline of Mainstream LLMs

We tested several state-of-the-art large language models, including GPT-4, Claude-3, LLAMA-3, and others, on our CausalBench. The evaluation metrics included: Four-Type Questions Group Correction Rate, Overall Correction Rate (Ignore Question Type), From Cause to Effect without Intervention Correction Rate, From Effect to Cause without Intervention Correction Rate, From Cause to Effect with Intervention Correction Rate, and From Effect to Cause with Intervention Correction Rate. For each causal scenario, there are four questions: cause-to-effect without intervention, effect-to-cause without intervention, cause-to-effect with intervention, and effect-to-cause with intervention. The Four-Type Questions Group Correction Rate represents the proportion of scenario cases where all four types of questions of one scenario are all answered correctly by the large language models. If any of the four questions of a scenario is answered incorrectly, the scenario is considered to be answered incorrectly by the LLM. The Overall Correction Rate (Ignore Question Type) is calculated by dividing the total number of correctly answered questions by the total number of questions, without categorizing the questions by type and scenario. The From Cause to Effect without Intervention Correction Rate is calculated by dividing the number of correctly answered "From Cause to Effect without Intervention" type questions by the total number of this type of questions. Similarly, the From Effect to Cause without Intervention Correction Rate is calculated by dividing the number of correctly answered "From Effect to Cause without Intervention" type questions by the total number of this type of questions. The remaining two metrics, From Cause to Effect with Intervention Correction Rate and From Effect to Cause with Intervention Correction Rate, follow the same calculation method as the previous two metrics, focusing on their respective question types.

Here are the tables showing LLMs' performance on text, math, and code problems.

3.2 Test Result Summary

The evaluation results of state-of-the-art large language models on CausalBench provide valuable insights into their causal reasoning capabilities across textual, mathematical, and coding problem domains:

Overall, the models achieved higher correction rates on mathematical problems compared to textual and coding problems. For instance, GPT-4 achieved an 88.7% overall correction rate on math problems, while scoring 73.3% and 71.0% on text and code problems, respectively. This suggests that causal reasoning in mathematical contexts is relatively easier for LLMs compared to natural language and programming domains.

The Four-Type Questions Group Correction Rate, which measures the proportion of scenarios where all four reasoning perspectives are correctly answered, was consistently lower than the Overall Correction Rate (Ignore Question Type) across all problem types. For example, GPT-4 achieved a 61.4% Four-Type Questions Group Correction Rate on math problems, compared to an 88.7% Overall Correction Rate. This indicates that LLMs often struggle to maintain a comprehensive understanding of causal relationships when questioned from multiple perspectives.

The introduction of interventions in the causal scenarios led to mixed results in correction rates across models and problem types. In the text domain, the correction rates slightly decreased for most models when interventions were introduced. However, in the math domain, the correction rates generally improved with interventions. For instance, GPT-4's performance increased from 78.6% to 91.7% on cause-to-effect questions with intervention in math problems. In the coding domain, the impact of interventions varied across models, with some showing improvements and others exhibiting a decline in performance.

Among the tested models, GPT-4 and Claude-3 consistently outperformed other large language models (LLMs) across most problem types and reasoning dimensions, achieving the highest cor-

Model	Four-Type Questions Group Correction Rate(%)	Overall Correction Rate(Ignore Question Type) (%)	From Cause to Effect without Intervention Correction Rate (%)	From Effect to Cause without Intervention Correction Rate (%)	From Cause to Effect with Intervention Correction Rate (%)	From Effect to Cause with Intervention Correction Rate (%)
GPT-4 Turbo	36.9	73.3	74.4	71.2	73.8	73.7
Claude3-Opus	36.8	72.6	74.1	70.9	73.2	72.2
Mistral-7B	25.5	63.6	58.7	66.5	64.2	65.0
Llama3-70B	21.8	61.5	62.6	59.6	63.8	60.1
Llama2-7B	20.7	62.1	62.8	64.0	56.4	65.4
GPT-3.5	16.7	57.8	57.6	58.5	56.2	58.7
Gemma-7b-it	12.8	50.7	50.0	46.9	53.6	52.1
Bloomz	4.2	41.7	41.0	40.7	41.7	43.6
AquilaChat	1.9	31.1	28.7	32.4	33.1	30.4

Table 1: LLM Performance on Text Problems.

Model	Four-Type Questions Group Correction Rate(%)	Overall Correction Rate(Ignore Question Type) (%)	From Cause to Effect without Intervention Correction Rate (%)	From Effect to Cause without Intervention Correction Rate (%)	From Cause to Effect with Intervention Correction Rate (%)	From Effect to Cause with Intervention Correction Rate (%)
Mistral-7B	62.0	87.2	78.9	85.6	85.3	98.9
GPT-4 Turbo	61.4	88.7	78.6	88.3	91.7	96.0
Claude3-Opus	54.6	85.9	74.7	87.1	86.5	95.4
Llama3-70B	40.8	80.7	56.8	86.8	82.0	97.1
Gemma-7b-it	38.3	79.2	50.4	82.8	91.1	92.0
AquilaChat	25.3	68.1	57.0	67.8	69.2	78.3
Bloomz	23.9	69.2	53.3	76.8	67.3	79.7
GPT-3.5	15.9	63.3	47.1	71.5	48.6	86.1
Llama2-7B	2.8	42.3	45.3	54.2	17.5	52.4

Table 2: LLM Performance on Problems.

Model	Four-Type Questions Group Correction Rate(%)	Overall Correction Rate(Ignore Question Type) (%)	From Cause to Effect without Intervention Correction Rate (%)	From Effect to Cause without Intervention Correction Rate (%)	From Cause to Effect with Intervention Correction Rate (%)	From Effect to Cause with Intervention Correction Rate (%)
Llama3-70B	43.8	77.0	82.0	75.7	73.9	76.0
Claude3-Opus	39.6	71.3	78.6	71.3	68.7	66.5
GPT-4	37.2	71.0	80.6	67.5	73.2	62.5
Gemma	32.3	68.4	74.1	67.7	66.0	65.4
Mistral	31.4	66.8	67.5	68.3	61.3	70.2
GPT-3.5	25.0	64.5	71.9	65.4	59.8	60.6
Llama2-7B	22.6	61.9	79.0	45.5	76.3	46.8
Bloomz	17.5	52.4	49.6	56.8	46.4	56.8
AquilaChat	14.7	47.3	36.8	56.4	38.9	57.2

Table 3: LLM Performance on Code Problems.

rection rates. Mistral demonstrated strong performance in mathematical problems but exhibited shortcomings in code-related tasks. Conversely, LLAMA-3 showed robust performance in code-related problems but faced challenges with text and mathematical tasks.

4 Correlation with Hallucination

To analyze the correlation between LLMs' causal reasoning ability and their hallucination rate, we referred to the LLMs' performance on hallucination datasets. The hallucination evaluation results were obtained from the Hallucination Leaderboard, developed by Vectara (Hughes and Bae, 2023). This leaderboard provides a comparison of LLM performance in maintaining a low hallucination rate and ensuring factual consistency when summarizing a set of facts.

The hallucination evaluation process involves measuring the hallucination rate, factual consistency rate, answer rate, and average summary length. These metrics provide a comprehensive understanding of each model's tendency to hallucinate and its ability to maintain factual accuracy (Hughes and Bae, 2023).

After comparing the LLMs' performance on CausalBench with their performance on the Hallucination evaluation leaderboard provided by Vectara on Huggingface (Hughes and Bae, 2023), we found that models with stronger causal reasoning abilities tend to exhibit lower hallucination rates. For instance, GPT-4 Turbo, LLAMA-3-70B, and Mistral-7B, which demonstrated superior performance on causal reasoning tasks, also had low hallucination rates. In contrast, models like Google Gemma-7b-it and LLAMA-2-7B, which showed weaker performance on our CausalBench, had higher hallucination rates of 7.5% and 5.6%,

Model	Hallucination Rate	Factual Consistency Rate	Answer Rate)	Average Summary Length (Words)
GPt-4 Turbo	2.5%	97.5%	100.0%	86.2
Llama3-70B	4.5%	95.5%	99.2%	68.5
Mistral 7B Instruct-v0.2	4.5%	95.5%	100.0%	106.1
Llama2-7B	5.6%	94.4%	99.6%	119.9
Claude3-Opus	7.4%	92.6%	95.5%	92.1
Google Gemma-7b-it	7.5%	92.5%	100.0%	113.0

Table 4: Performance of LLMs on the Hallucination Dataset.

respectively.

This trend indicates a potential link between a model's ability to understand and reason about causal relationships and its likelihood of not producing hallucinations. Further research is required to explore this correlation in more depth and to understand the underlying mechanisms driving this relationship.

5 Impact and Limitations

5.1 Impact

For the first time, we innovatively propose four types of questioning approaches for the same causal scenario: cause-to-effect, effect-to-cause, cause-to-effect with intervention, and effect-to-cause with intervention. We also calculate the proportion of cases where large language models correctly answer all four types of questions for a given causal scenario. This effectively avoids the situation where large language models coincidentally answer causal questions correctly without understanding the causal relationships embedded in the causal scenario, thereby improving the accuracy of the dataset's test results. By providing causal reasoning problems spanning multiple domains(text, code, math), it addresses the limitations of existing causal datasets and offers a more comprehensive and robust tool for assessing the causal reasoning abilities of language models. The findings in this paper suggest that models with stronger causal reasoning capabilities tend to exhibit lower hallucination rates, providing a new perspective on exploring the relationship between causal reasoning and reducing hallucinations. CausalBench has the potential to become a benchmark for driving progress in causal reasoning in artificial intelligence.

6 Conclusion

In this paper, we present CausalBench, a comprehensive benchmark dataset for evaluating the causal reasoning capabilities of large language models. CausalBench innovatively proposes four types of questioning approaches for each causal scenario: cause-to-effect, effect-to-cause, cause-to-effect with intervention, and effect-to-cause with intervention. By calculating the proportion of cases where models correctly answer all four question types, CausalBench effectively assesses whether LLMs truly understand the underlying causal relationships, mitigating the impact of models coincidentally providing correct answers without causal comprehension.

The dataset encompasses a diverse set of problems spanning textual, mathematical, and coding domains, addressing the limitations of existing causal reasoning benchmarks. Evaluated on CausalBench, state-of-the-art LLMs demonstrate stronger performance on mathematical problems compared to textual and coding tasks. Notably, models with superior causal reasoning abilities tend to exhibit lower hallucination rates, suggesting a potential link between the two capabilities.

Despite its contributions, CausalBench has several limitations, including the need for expanded domain coverage and deeper exploration of the intrinsic mechanisms connecting causal reasoning and hallucination reduction. Future work will focus on addressing these limitations, further refining the evaluation metrics, and providing insights to advance the development of causal reasoning abilities in large language models. CausalBench serves as a robust tool and an important step towards achieving artificial general intelligence.

Limitations

CausalBench has several limitations that need to be addressed in future work. These include the need for further expanding the domain coverage, increasing the scale of the dataset, incorporating causal discovery tasks and exploring the intrinsic mechanisms between causal reasoning and hallucinations through more empirical studies.

References

Frances E. Allen. 1970. Control flow analysis. *ACM SIGPLAN Notices*, 5(7):1–19.

Joshua D. Angrist, Guido W. Imbens, and Donald B. Rubin. 1996. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91(434):444–455.

Leshem Choshen, Paarth Neekhara, Kyle Richardson, Lisa Xue, Madian Hou, Shehzaad Neekhara, Yao Chen, and Heike Adel. 2022. Cladder: A causal language model for causal reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6205–6224.

Ron Cytron, Jeanne Ferrante, Barry K. Rosen, Mark N. Wegman, and F. Kenneth Zadeck. 1991. Efficiently computing static single assignment form and the control dependence graph. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 13(4):451–490.

Jeanne Ferrante, Karl J. Ottenstein, and Joe D. Warren. 1987. The program dependence graph and its use in optimization. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 9(3):319–349.

Lex Fridman and Judea Pearl. 2022. Causal reasoning, counterfactuals, and the path to agi. *Miniature Brain Machinery Webinar Review*.

Andreas Griewank and Andrea Walther. 2008. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM.

Mary Jean Harrold and Gregg Rothermel. 1994. Performing data flow testing on classes. In *Proceedings of the 2nd ACM SIGSOFT Symposium on Foundations of Software Engineering (SIGSOFT'94)*, pages 154–163.

Simon Hughes and Minseok Bae. 2023. Vectara hallucination leaderboard. https://github.com/vectara/hallucination-leaderboard.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona T. Diab, and Bernhard Schölkopf. 2023. Can large language models infer causation from correlation? *CoRR*, abs/2306.05836.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430.

Pengfei Li and Kezhi Mao. 2019. Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts. *Expert Systems with Applications*, 115:512–523.

Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, et al. 2017. Sympy: Symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Joel Moses. 1971. Algebraic simplification: A guide for the perplexed. *Communications of the ACM*, 14(8):527–537.

Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688.

Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Judea Pearl et al. 2000. *Causality: Models, reasoning and inference*, volume 29. Cambridge university press.

Python Software Foundation. 2023. ast — abstract syntax trees. https://docs.python.org/3/library/ast.html. Accessed: 2023-06-05.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 658–666.