Answering Causal Questions with Augmented LLMs

Nick Pawlowski [1] **James Vaughan** [1] **Joel Jennings** [1] **Cheng Zhang** [1]

Abstract

Large Language Models (LLMs) are revolutionising the way we interact with machines, enabling entirely new applications. An emerging use case for LLMs is to provide a chat interface to complex underlying systems, allowing natural language interaction without the need for the user to learn system specifics. This also allows LLMs to be augmented to perform tasks that they are ill-suited to perform by themselves. One example of this is precise causal reasoning. In this paper, we explore one component in building conversational systems with causal question-answering capabilities. Specifically, we augment LLMs with access to precomputed outputs of a causal expert model to examine their effectiveness at answering causal questions by providing either: 1) the predicted causal graph and related treatment effects to the LLM context; 2) access to an API to derive insights from the output of the causal model. Our experiments show that neither method is able to fully solve the task. However, context-augmented LLMs make significantly more mistakes than the data-access API-augmented LLMs, which are invariant to the size of the causal problem. We believe that the insights generalize to complex reasoning tasks beyond causal reasoning and we hope to inspire further research into building causality-enabled conversational systems.

1. Introduction

In the recent years, the field of LLMs (OpenAI, 2023; Thoppilan et al., 2022) has evolved from research to driving the growth of businesses as well as leading to the inception of previously impossible applications, assisting users in solving programming, text editing or data modelling tasks.

However, LLMs have not yet been used to aid in causal data analysis, such as causal effect inference or causal discovery. Generally, LLMs are not designed to do causal reasoning (Zhang et al., 2023) despite being able to summarize known causal knowledge (Kıcıman et al., 2023). The core of causal reasoning consists of building a model of the underlying true data generating mechanism to reason about the consequences of any intervention. LLMs on the other hand are restricted to answering questions about the common knowledge contained in its training data or the causality in a given text (Kıcıman et al., 2023).

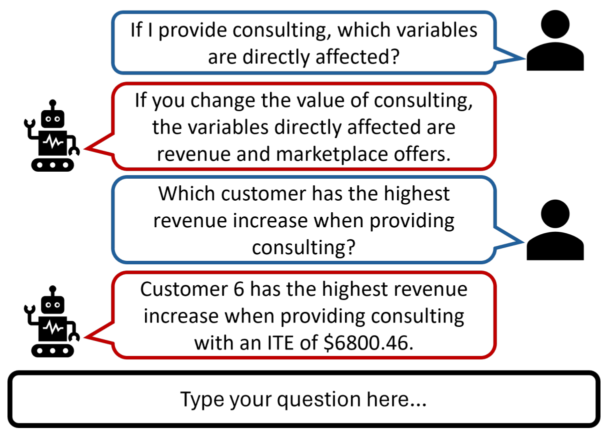

Figure 1. A causality-augmented LLM enables the answering of questions about the relationships of variables as well as support individual decision making.

The aim of this paper is to take a first step to imbuing LLMs with the capabilities to answer causal queries about a given dataset such as depicted in Figure 1. A fully causality-enabled LLM requires a causal module that is sufficiently accurate and generally applicable to answer questions about any potential query dataset. Instead of building the full system, we tackle the simplified problem of answering questions about the causal information extracted from a dataset using an expert system and let future work tackle the integration of this expert system[1] into an LLM. The general idea to augment LLMs is not new (OpenAI, 2023; Schick et al., 2023; Mialon et al., 2023; Zhang, 2023). We focus on the question of how to provide the LLM access to the necessary

[1]Microsoft Research, Cambridge, UK. Correspondence to: Nick Pawlowski <nick.pawlowski@microsoft.com>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

[1]Expert systems could be build with packages such as EconML (Battocchi et al., 2019) or DECI (Geffner et al., 2022).

information and evaluate the feasibility and limitations of today's LLMs to answer causal questions. We conclude by discussing the future directions for LLMs, using causal reasoning as an example of a task that LLMs alone cannot currently solve.

Our contributions are:

- We compare methods for allowing an LLM to answer causal questions by providing it access to information from a causal expert system in two ways: 1) via context augmentation: appending it to context of the prompt and 2) via tool augmentation: providing a simple API for manipulating the expert system's output.

- We evaluate the effectiveness of our proposed methods on a range of synthetic questions about the causal graph or the predicted treatment effects.

- We show that LLMs alone are incapable of answering causal question with context augmentation alone, but require access to additional tools for reliable performance.

2. Related work

With the recent success of LLMs, research on building applications with them has been growing. A lot of work has focused on expanding an LLM's capabilities by providing access to specialised tools (OpenAI, 2023; Schick et al., 2023; Mialon et al., 2023; Zhang, 2023), also called augmented language models (ALMs). These tools included features such as information retrieval and calculators. The tool usage can be learned, fine-tuned or more simply through in-context learning. In-context learning allows for the integration of new plugins into a trained LLM and the principle underlies the ChatGPT plugin system. Similarly, in-context learning has been used to improve the reasoning capabilities of LLMs by teaching a new skill through few-shot examples within the prompt (Zhou et al., 2022).

Another line of related work explores the usage of LLMs for causal reasoning tasks. Zhang et al. (2023) argues that LLMs are not yet capable of performing certain causal tasks such as causal discovery, but can be extended through the right augmentation. On the other hand, it has been shown that LLMs can access common knowledge to solve non-data driven causal discovery (Gao et al., 2023; Tu et al., 2023) as well as other causal tasks (Kıcıman et al., 2023). However, the solution of those tasks depend on the knowledge encoded in the LLM rather than performing data-driven causal modelling. As such, LLMs are incapable of extracting novel causal insights from previously unseen data (Zhang et al., 2023).

3. Methodology

In this work, we assume that there is a causal expert system that provides causal insights given some observational data. Specifically, it gives access to a causal graph as well as a list of predicted individual treatment effects. We choose to give these two types of information as this is the typical output of causal expert system using existing causal discovery (Glymour et al., 2019), causal inference (Pearl, 2010; Battocchi et al., 2019) or end-to-end causal inference methods (Geffner et al., 2022). We are then interested in using LLMs to answer questions about the causal relations encoded in the causal graph, the optimal treatment assignment given the individual treatment effects, and the average treatment effects implicit in the list of individual treatment effects.

Our method is based on simple text completion and we consider two different augmentation approaches to allow LLMs to answer these questions: 1) context augmentation, which represents the scenario where LLMs are used for all additional operations beyond the causal expert system and 2) tool augmentation, where additional basic python tools can be used to manipulate the causal expert system's output to transform into an easier to parse format for the LLM.

3.1. Context Augmentation

The naive approach of giving LLMs causal capabilities relies on providing the graph as well as the ITE table in the prompt, together with the question to be answered. In practice, we found that this method is very limited by the context length and instead we implemented a two-stage approach that first classifies whether a question is related to the graph or to the ITE table. The second stage prompt is then tailored to the specific question type and only contains the relevant causal information from which to extract the answer.

We implemented this with three different prompts. First, classify a given question (see Listing 1). This prompt combines chain-of-thought reasoning with in-context learning. Specifically, we gave examples of graph-type questions as well as treatment effect-related questions. Finally, we asked the LLM to output its reasoning as well as the classification in a JSON format.

Depending on the classification result, we used a specific prompt template for either graph-related questions (see Listing 2) or treatment effect-related questions (see Listing 3). The graph template provides information about the causal graph in GraphML format and adds an example parent-child relationship from the causal graph. The treatment effect template provides the ITE table in CSV format. We used GraphML and CSV as these formats are commonly used to represent graphs and tables. Both templates ask the LLM to perform step-by-step reasoning and output the intermediate steps, as well as providing the result in JSON format.

3.2. Tool Augmentation

Our second approach augmented the LLM with a custom API that allows it to request the output to a chain of different operations by generating a JSON list with the corresponding API calls. The LLM receives the output in JSON format and uses it to compose the final response. In practice, we implemented this approach using two completion requests, the first to generate the API call and the second to generate the final response based on the injected output of the API call.

The prompt is shown in Listing 4, it contains a description of the available functions that the LLM has access to as well as an example of the general behaviour of the API. Additionally, the prompt provides some examples of graph, average treatment effect and individual treatment effect questions together with the corresponding API calls and outputs. Finally, the model is asked to think step by step to answer the user's question.

3.3. JSON parsing

In our initial experiments, we found that the LLM would often output more than just a single JSON object. To deal with this, we attempted to parse the content between the initial prompted open (curly or square) bracket and the last occurrence of the closing equivalent. In case of failure, we report the attempt to be erroneous.

4. Experiments and Results

Method n_n	Tool	Context
5	0.16 ± 0.37	0.00 ± 0.00
10	0.10 ± 0.30	0.00 ± 0.00
20	0.02 ± 0.14	0.01 ± 0.10
40	0.03 ± 0.17	0.03 ± 0.16

Table 1. Proportion of responses to graph-type questions that were not given as parse-able JSON objects. n_n are the number of nodes in the graph.

Method n_n	Tool	Context
5	0.57 ± 0.49	0.16 ± 0.37
10	0.82 ± 0.38	0.06 ± 0.24
20	0.90 ± 0.30	0.09 ± 0.29
40	0.93 ± 0.26	0.08 ± 0.27

Table 2. Correctness of the responses to graph-type questions that were given as parse-able JSON objects. n_n are the number of nodes in the graph.

 n_t	Method n_s	Tool	Context
5	10	0.48 ± 0.50	0.18 ± 0.38
	30	0.36 ± 0.48	0.14 ± 0.35
	100	0.40 ± 0.49	0.24 ± 0.43
10	10	0.60 ± 0.49	0.30 ± 0.46
	30	0.60 ± 0.49	0.28 ± 0.45
	100	0.54 ± 0.50	1.00 ± 0.00
20	10	0.60 ± 0.49	0.14 ± 0.35
	30	0.56 ± 0.50	0.00 ± 0.00
	100	0.52 ± 0.50	1.00 ± 0.00
40	10	0.60 ± 0.49	0.08 ± 0.27
	30	0.60 ± 0.49	1.00 ± 0.00
	100	0.60 ± 0.49	1.00 ± 0.00

Table 3. Proportion of responses to treatment effect questions that were *not* parse-able as JSON objects. n_t refers to the number of available treatment variables. n_s are the number of individual subjects in each table.

We evaluated the performance of the augmented LLMs in terms of the syntactic correctness of the text completion, i.e. whether we can parse the JSON object, and then test the final response for its factual correctness. We simulated the practical application of LLMs answering causal questions by synthetically generating causal graphs as well as ITE tables for different sized causal problems. In practice, these graphs and treatment effect estimates will be the output of a causal expert system. We generated questions about the causal graph and the ITE table and asked the LLMs to answer them. We evaluated the final answer by letting another LLM compare it to an ideal answer for the given question by adapting the `closed_qa` model-graded evaluation template from the OpenAI evals package as shown in Listing 5. We evaluated the performance of the LLMs on the generated questions and report the results in Tables 1 to 4.

We generated random Erdos-Renyi graphs with $n_n \in [5, 10, 20, 40]$ nodes and use an edge existence probability of $p = 0.2$. The ITE tables show the individual treatment effect of $n_t \in [5, 10, 20, 40]$ on $n_s \in [10, 30, 100]$ subjects. Additionally, we provided a column showing whether the subject has already been treated or not. The treatment effects were randomly sampled as $e = \epsilon * 100 - 30 \mid \epsilon \sim \mathcal{N}(0, 1)$. Overall, we ran every setup with 5 different random seeds and averaged the performance.

The questions are generated using specific templates. We have four different templates for graph type questions and seven different templates for treatment effect type questions. The graph type questions are:

- **(Connectivity)** Does a change in X lead to a change in Y?

	Method	Tool	Context
n_t	n_s		
5	10	0.69 ± 0.46	0.24 ± 0.43
	30	0.47 ± 0.50	0.21 ± 0.41
	100	0.47 ± 0.50	0.26 ± 0.44
10	10	0.75 ± 0.43	0.14 ± 0.35
	30	0.75 ± 0.43	0.14 ± 0.35
	100	0.48 ± 0.50	-
20	10	0.55 ± 0.50	0.09 ± 0.29
	30	0.73 ± 0.45	0.04 ± 0.20
	100	0.67 ± 0.47	-
40	10	0.80 ± 0.40	0.04 ± 0.20
	30	0.70 ± 0.46	-
	100	0.80 ± 0.40	-

Table 4. Correctness of responses to treatment effect questions that were parse-able as JSON objects. n_t refers to the number of available treatment variables. n_s are the number of individual subjects in each table.

- **(Paths)** Through which variables does X influence Y?
- **(Direct Parents)** Which variables directly influence X?
- **(Direct Children)** If I change the value of X, which variables are directly affected?

The treatment effect-type questions all ask about the optimal treatment for either the full table or under some condition or a combination thereof: 1) not yet being engaged; 2) for a specific treatment; 3) for a specific subject/patient.

We evaluate the effectiveness of both augmentation approaches with the Azure OpenAI Service API using GPT-3.5 Turbo. We set the maximum response length to 2000 tokens and use a temperature of $t = 10^{-10}$. We show additional initial GPT-4 results in the appendix in Appendix A.

4.1. Quantitative Results

Comparing the performance of the context-augmented approach to the tool-augmented approach on the graph questions, we see that the context-augmented method shows very few errors in generating parse-able JSON outputs, while the tool-based prompt sometimes generates invalid JSON outputs or API prompts. However, the correctly parse-able responses of the tool-based prompt are more often correct than the context-augmented prompt, where the tool-based approach achieves almost perfect performance on the larger graphs. This is likely due to the fact that the tool-augmented method does not require the LLM to reason over the graph, but can use the API to do the reasoning on its behalf. Additionally, both methods show a different trend in the performance as a function of the size of the graph. The results of the context-augmented approach get worse with a bigger graph, while the tool-augmented method results get better. This is likely due to the fact that the context-augmented approach needs to perform more reasoning steps to answer the question with an increasing graph size, while the tool-augmented method can use the same API calls for the same question but different graph sizes.

In the case of treatment effect-type questions, we find that both methods degrade in performance, especially the JSON formatting, when the size increases. This could be explained by the the increase in context where some combinations always fail because the added ITE CSV extends beyond the maximal context length. The tool-based method is relatively robust to the problem size because the context size is near-constant between all settings. On the other hand, context-augmentation depends on the full table within the context and can therefore only solve a limited number of problems.

4.2. Qualitative Results

Inline with our quantitative findings, the tool-augmented approach seems to provide higher quality responses in terms of language as well. This is consistent with the intuition that the context-augmented method has to perform the actual reasoning, whereas the tool-augmented method can focus more on the language modelling task. Generally, the context-augmented answers are rather short and concise. As one can see in the examples in 5.

5. Discussion & Conclusion

In this paper, we show that augmented LLMs are capable of answering causal questions assuming that it is provided the necessary information from a causal expert system. We propose two different methods to augmenting an LLM with the required information: 1) The context-augmented approach provides the expert system data within the prompt that is used to answer the question. The performance of this method deteriorates with increasing problem sizes with the context size also increasing. Furthermore, the LLM struggles with abstract reasoning and often provides sensible but incorrect answers. 2) The tool-based approach provides access to the expert system data via an API that performs the relevant reasoning tasks. This method is more robust to increasing problem sizes and achieves significantly better performance than the context-augmentation approach. This is likely because the tool-based prompt abstracts the reasoning task away from the LLM and therefore allows it to focus on the language modeling task. In general this is inline with previous works on augmented LLMs and demonstrates the effectiveness of providing the correct tools to LLMs (Mialon et al., 2023).

However, our proposed method does not perfectly solve the

Question	Context	Tool	Ideal
If I change the value of X3, which variables are directly affected?	✗ X4	✗ Sorry, I cannot answer this question as there is no information about the relationship between X3 and other variables in the causal graph.	X3 does not directly influence any variables
Which variables do directly influence X3?	✔ X0	✔ The variable that directly influences X3 is X0.	X0 directly influence X3
Through which variables does X1 influence X2?	✗ X3	✗ There is no direct path between X1 and X2 in the causal graph. Would you like me to check for indirect paths?	X1 does not influence X2
Does a change in X0 lead to a change in X2?	✗ Yes, a change in X0 leads to a change in X2 according to the given causal graph.	✔ No, there is no path between 'X0' and 'X2' in the causal graph, so a change in 'X0' does not lead to a change in 'X2'.	X0 does not influence X2
Which variables do directly influence X0?	✔ X2	✗ JSON error	X2 directly influence X0
If I change the value of X0, which variables are directly affected?	✗ Variables X1 and X2 are directly affected by changing the value of X0.	✔ I'm sorry, but I cannot answer that question without knowing the causal relationships between X0 and the other variables in the graph. Can you provide more information about the causal graph?	X0 does not directly influence any variables

Table 5. Qualitative Examples of correct and incorrect answers of the two methods. The last row is an example of wrong grading by the grading LLM in case of the tool-augmented method.

tasks and further research is still needed. Our analysis is restricted by the use of GPT-3.5 Turbo instead of using the more powerful GPT-4 (OpenAI, 2023) model. This leads to imperfect JSON formatting causing parsing issues as well as problems with reasoning. Our first experiments with GPT-4 (see Appendix A) indicate that the use of GPT-4 improves the performance. However, the tool-augmented GPT-4 seems to output additional text and JSON objects beyond the original completion which requires tweaking of the JSON parsing. Additionally, we find that the LLM dependent model evaluation is not perfect and sometimes leads to false negatives - responses that are graded correct even though they are incorrect and vice-versa. A more elaborate study could rely on human annotation to solve this problem.

Overall, this work is only a first step in enabling LLMs to answer causal questions or more general complex reasoning tasks. Building a fully causally-enabled LLM model requires building and integrating the currently assumed expert system. This system could be based on packages such as DECI (Geffner et al., 2022) or EconML (Battocchi et al., 2019). Future research should implement an API to call this expert system and experiment with different prompts based on our tool-augmented LLM. However, our study suggests that simple in-context-learned tool-augmentation might not be enough for LLMs to solve this task and more elaborate prompts and JSON parsing might be needed for satisfying performance. Nevertheless, this is a first step towards building causality-enabled conversational systems that could revolutionise how decisions are made.

References

Battocchi, K., Dillon, E., Hei, M., Lewis, G., Oka, P., Oprescu, M., and Syrgkanis, V. Econml: A python package for ml-based heterogeneous treatment effects estimation. *Version 0. x*, 2019.

Gao, J., Ding, X., Qin, B., and Liu, T. Is chatgpt a good causal reasoner? a comprehensive evaluation. *arXiv preprint arXiv:2305.07375*, 2023.

Geffner, T., Antoran, J., Foster, A., Gong, W., Ma, C., Kiciman, E., Sharma, A., Lamb, A., Kukla, M., Pawlowski, N., et al. Deep end-to-end causal inference. *arXiv preprint arXiv:2202.02195*, 2022.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Kıcıman, E., Ness, R., Sharma, A., and Tan, C. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Rozière, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A., et al. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*, 2023.

OpenAI. GPT-4 technical report. March 2023.

OpenAI. Introducing plugins for chatgpt. https://openai.com/blog/introducing-plugins-for-chatgpt/, July 2023.

Pearl, J. Causal inference. *Causality: objectives and assessment*, pp. 39–58, 2010.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Tu, R., Ma, C., and Zhang, C. Causal-discovery performance of chatgpt in the context of neuropathic pain diagnosis. *arXiv preprint arXiv:2301.13819*, 2023.

Zhang, C., Bauer, S., Bennett, P., Gao, J., Gong, W., Hilmkil, A., Jennings, J., Ma, C., Minka, T., Pawlowski, N., et al. Understanding causality with large language models: Feasibility and opportunities. *arXiv preprint arXiv:2304.05524*, 2023.

Zhang, J. Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. *arXiv preprint arXiv:2304.11116*, 2023.

Zhou, H., Nova, A., Larochelle, H., Courville, A., Neyshabur, B., and Sedghi, H. Teaching algorithmic reasoning via in-context learning. *arXiv preprint arXiv:2211.09066*, 2022.

A. Additional Results

	GPT-3.5 Turbo-chat		*GPT-4*	
Method	Tool	Context	Tool	Context
n_n				
5	0.16 ± 0.37	0.00 ± 0.00	0.44 ± 0.50	0.00 ± 0.00
10	0.10 ± 0.30	0.00 ± 0.00	0.52 ± 0.50	0.00 ± 0.00
20	0.02 ± 0.14	0.01 ± 0.10	0.57 ± 0.50	0.00 ± 0.00
40	0.03 ± 0.17	0.03 ± 0.16	0.41 ± 0.49	0.00 ± 0.00

Table 6. Proportion of responses to graph-type questions that were not given as parse-able JSON objects. n_n are the number of nodes in the graph.

	GPT-3.5 Turbo-chat		*GPT-4*	
Method	Tool	Context	Tool	Context
n_n				
5	0.57 ± 0.49	0.16 ± 0.37	1.00 ± 0.00	0.20 ± 0.40
10	0.82 ± 0.38	0.06 ± 0.24	1.00 ± 0.00	0.08 ± 0.27
20	0.90 ± 0.30	0.09 ± 0.29	1.00 ± 0.00	0.02 ± 0.14
40	0.93 ± 0.26	0.08 ± 0.27	1.00 ± 0.00	0.01 ± 0.12

Table 7. Correctness of the responses to graph-type questions that were given as parse-able JSON objects for *GPT-3.5 Turbo* and *GPT-4*. n_n are the number of nodes in the graph.

		GPT-3.5 Turbo-chat		*GPT-4*	
	Method	Tool	Context	Tool	Context
n_t	n_s				
5	10	0.48 ± 0.50	0.18 ± 0.38	0.98 ± 0.14	0.00 ± 0.00
	30	0.36 ± 0.48	0.14 ± 0.35	0.96 ± 0.20	0.00 ± 0.00
	100	0.40 ± 0.49	0.24 ± 0.43	0.92 ± 0.27	0.00 ± 0.00
10	10	0.60 ± 0.49	0.30 ± 0.46	0.84 ± 0.37	0.00 ± 0.00
	30	0.60 ± 0.49	0.28 ± 0.45	0.92 ± 0.27	0.00 ± 0.00
	100	0.54 ± 0.50	1.00 ± 0.00	0.84 ± 0.37	1.00 ± 0.00
20	10	0.60 ± 0.49	0.14 ± 0.35	0.82 ± 0.38	0.00 ± 0.00
	30	0.56 ± 0.50	0.00 ± 0.00	0.90 ± 0.30	0.00 ± 0.00
	100	0.52 ± 0.50	1.00 ± 0.00	0.76 ± 0.43	1.00 ± 0.00
40	10	0.60 ± 0.49	0.08 ± 0.27	0.76 ± 0.43	0.00 ± 0.00
	30	0.60 ± 0.49	1.00 ± 0.00	0.74 ± 0.44	1.00 ± 0.00
	100	0.60 ± 0.49	1.00 ± 0.00	0.86 ± 0.35	1.00 ± 0.00

Table 8. Proportion of responses to treatment effect questions that were *not* parse-able as JSON objects for *GPT-3.5 Turbo* and *GPT-4*. n_t refers to the number of available treatment variables. n_s are the number of individual subjects in each table. Generally, *GPT-3.5 Turbo* creates more parseable outputs for the tool-usage, whereas *GPT-4* provides valid JSON for most context augmentation. In case of tool usage, *GPT-4* often outputs additional text outside of valid JSON which interferes with our rudimentary parsing strategy.

		GPT-3.5 Turbo-chat		*GPT-4*	
	method	Tool	Context	Tool	Context
n_t	n_s				
5	10	0.69 ± 0.46	0.24 ± 0.43	1.00 ± 0.00	0.48 ± 0.50
	30	0.47 ± 0.50	0.21 ± 0.41	1.00 ± 0.00	0.42 ± 0.49
	100	0.47 ± 0.50	0.26 ± 0.44	1.00 ± 0.00	0.22 ± 0.41
10	10	0.75 ± 0.43	0.14 ± 0.35	1.00 ± 0.00	0.30 ± 0.46
	30	0.75 ± 0.43	0.14 ± 0.35	1.00 ± 0.00	0.28 ± 0.45
	100	0.48 ± 0.50	-	1.00 ± 0.00	-
20	10	0.55 ± 0.50	0.09 ± 0.29	1.00 ± 0.00	0.26 ± 0.44
	30	0.73 ± 0.45	0.04 ± 0.20	1.00 ± 0.00	0.12 ± 0.32
	100	0.67 ± 0.47	-	1.00 ± 0.00	-
40	10	0.80 ± 0.40	0.04 ± 0.20	1.00 ± 0.00	0.18 ± 0.38
	30	0.70 ± 0.46	-	1.00 ± 0.00	-
	100	0.80 ± 0.40	-	0.86 ± 0.35	-

Table 9. Correctness of responses to treatment effect questions that were parse-able as JSON objects for both *GPT-3.5 Turbo* and *GPT-4*. n_t refers to the number of available treatment variables. n_s are the number of individual subjects in each table. Empty fields correspond to no JSON parse-able responses.

B. Prompts

```
You are a personal AI assistant that can answer causal questions about some data. You can either answer questions about the causal graph, average treatment effects (ATEs) or individual treatment effects (ITEs). Questions about the graph deal with the relationship between different variables in the graph. Questions about the ATE care about the average outcomes of interventions or engagements. Questions about the ITE deal with individual outcomes that are on the level of a specific partner.

Here are some examples of each.
Graph Questions:
- List all paths between ABC and DEF.
- Is A a cause of B?
- Is C a parent of H?
- Is G an ancestor of B?

TE Questions:
- On average, whats the effect of the engagement "D"?
- When engaging with the program H, what outcome can we expect?
- When applying the engagement "L" on partner 1, what's the effect?
- Which partner has the highest revenue growth when providing engagement C?

Classify the question you receive into one of these two categories. Provide a reasoning for the classification.

Only output the results in the following JSON format:
{
"class": "",
"reason": "",
}

For example:

{
"class": "graph",
"reason": "The question is about the causal graph."
}

The class can only be one of "graph", "te".

Question:
$question

{
```

Listing 1. Question Classification Template

```
Here is a causal graph as a  GraphML string.

$graph

In this example, $example_parent is a cause of $example_child.

There will be a question about the graph. What are the intermediate steps and solutions to them necessary to answer the question? Be as detailed as possible.
Make sure to consider all paths of any length when you are asked about paths in the graph, ie you should consider paths of length 1, 2, 3, 4, etc....

Also, provide an output on whether you believe the answer is correct or not.

Output your reasoning in the following JSON format:
{
  "steps": [
     {
      "Step": "",
```

```
      "Solution": ""
      }
  ]
  "answer": "",
  "correct":
}

Question:
$question

{
```

Listing 2. Graph Question Template

```
    There is a csv table holding the individual treatment effects (ITE) of different
    intervention (columns) on the revenue of a specific customer (rows).

There will be a question about the ITEs or average treatment effect (ATE) in the CSV. What
     are the intermediate steps and solutions to them necessary to answer the question? Be
     as detailed as possible.
Only output the results of the steps and the final results in the following JSON schema:
{
    "steps": [
        {"step": "", "result": ""}
    ],
    "answer": "",
}

For example:
{ "steps": [ {"step": "What is the average ITE of the Tech Support intervention?", "result
    ": "1234"} ],
"answer": "1234"
}

Here is the table:
$table

Question:
$question

{
```

Listing 3. Treatment Effect Template

```
    You are a personal AI assistant (CLLM) that can answer causal questions about some
    data. You can either answer questions about the causal graph, average treatment
    effects (ATEs) or individual treatment effects (ITEs). Questions about the graph deal
    with the relationship between different variables in the graph. Questions about the
    ATE care about the average outcomes of interventions or engagements. Questions about
    the ITE deal with individual outcomes that are on the level of a specific partner.

    Here are some examples of each.
    Graph Questions:
    - List all paths between ABC and DEF.
    - Is A a cause of B?
    - Is C a parent of H?
    - Is G an ancestor of B?

    ATE Questions:
    - On average, whats the effect of the engagement "D"?
    - When engaging with the program H, what outcome can we expect?

    ITE Questions:
    - When applying the engagement "L" on partner 1, what's the effect?
```

```
- Which partner has the highest revenue growth when providing engagement C?

You have access to the following data:
- A causal graph that describes the relationship between different variables. There are the following nodes in the graph:
$nodes
- A dataset that contains the outcomes of different engagements on different partners. This is a table with each row being a different partner and the following columns:
$interventions
    - `Already Engaged`: Whether the partner is already engaged with a program.

Additionally, you have access to a 'graph' API as well as a 'data' API. The graph API allows you to query the causal graph. The data API allows you to query the dataset.

The graph API provides the following functions:
- `get_variables()`: Returns all engagements.
- `get_paths_between(start, end)`: Returns all paths between the start and end node.
- `get_parents(node)`: Returns all direct parents of a node.
- `get_children(node)`: Returns all direct children of a node.
- `get_ancestors(node)`: Returns all ancestors of a node.
- `get_descendants(node)`: Returns all descendants of a node.

The data API allows you to interact with the dataset in Pandas DataFrame format. It provides the following functions:
- `get_data()`: Returns the entire dataset as CSV.
- `get_length()`: Returns the number of rows in the dataset.
- `index(row, column)`: Returns the value at the given row and column.
- `index(null, column)`: Returns the entire column.
- `index(row, null)`: Returns the entire row.
- `mean()`: Returns the mean of the entire dataset.
- `mean("columns")`: Returns the mean over the columns.
- `mean("rows")`: Returns the mean over the rows.
- `max()`: Returns the max of the entire dataset.
- `max("columns")`: Returns the max over the columns.
- `max("rows")`: Returns the max over the rows.
- `mask(condition)`: Returns the rows where the condition is true.

Additionally, you can pipe these functions together. For example, you can get the mean of the column "Tech Support" by calling `data.mean("columns").index(None, "Tech Support")`.

You can call these functions by using the following syntax:
[{"api_call": "graph.get_variables", "args": []}]
[{"api_call": "data.get_data", "args": []}]

And will get responses of the form
[{"api_call": "graph.get_variables", "args": []}, {"result": ["Tech Support", "Discount", "New Engagement Strategy", "Already Engaged"]}]
[{"api_call": "data.get_data", "args": []}, {"result": "Partner ID,Tech Support,Discount,New Engagement Strategy,Already Engaged\n1,0.1,0.2,0.3,True\n2,0.4,0.5,0.6,False\n3,0.7,0.8,0.9,True"}]

Some example queries and solutions look like this:

1) ATE question:
User: What's the average effect of the engagement "Tech Support"?
CLLM: [{"api_call": "data.mean", "args": ["rows"]}, {"api_call": "data.index", "args": [null, "Tech Support"]}, {"result": 0.4}, {"response": "The average effect of the engagement 'Tech Support' is 0.4."}]

2) ATE question:
User: What's the engagement with the highest average effect?
CLLM: [{"api_call": "data.mean", "args": ["rows"]}, {"api_call": "data.max", "args": []}, {"result": {"value": 0.6, "arg": "New Engagement Strategy"}}, {"response": "The engagement with the highest average effect is 'New Engagement Strategy' with an ATE of
```

```
 0.6."}]

3) ITE question:
User: Which partner has the highest revenue growth when providing engagement "Discount
"?
CLLM: [{"api_call": "data.index", "args": [bull, "Discount"]}, {"api_call": "data.max
", "args": []}, {"result": {"value": 0.8, "arg": 3}}, {"response": "Partner 3 has the
highest revenue growth when providing engagement 'Discount' with an ITE of 0.8."}]

4) ITE question:
User: When applying the engagement "New Engagement Strategy" on partner 2, what's the
effect?
CLLM: [{"api_call": "data.index", "args": [2, "New Engagement Strategy"]}, {"result":
0.6}, {"response": "When applying the engagement 'New Engagement Strategy' on partner
2, the effect is 0.6."}]

5) ITE question:
User: Which partner that has not yet been engaged with a program has the highest
revenue growth when providing engagement "Tech Support"?
CLLM: [{"api_call": "data.mask", "args": ["Already Engaged", false]}, {"api_call": "
data.index", "args": [null, "Tech Support"]}, {"api_call": "data.max", "args": []}, {"
result": {"value": 0.7, "arg": 3}}, {"response": "Partner 3 that has not yet been
engaged with a program has the highest revenue growth when providing engagement 'Tech
Support' with an ITE of 0.7."}]

6) Graph question:
User: Is "Tech Support" a cause of "Discount"?
CLLM: [{"api_call": "graph.get_paths_between", "args": ["Tech Support", "Discount"]},
{"result": [["Tech Support", "Discount"]]}, {"response": "Yes, 'Tech Support' is a
cause of 'Discount'."}]

7) Graph question:
User: Is "Discount" a parent of "Tech Support"?
CLLM: [{"api_call": "graph.get_parents", "args": ["Tech Support"]}, {"result": []}, {"
response": "No, 'Discount' is not a parent of 'Tech Support'."}]

8) Graph question:
User: Which variables influence "Revenue"?
CLLM: [{"api_call": "graph.get_ancestors", "args": ["Revenue"]}, {"result": ["Tech
Support", "Discount", "New Engagement Strategy"]}, {"response": "The variables that
influence 'Revenue' are 'Tech Support', 'Discount', and 'New Engagement Strategy'."}]

9) Out of context question:
User: What's your favourite pizza topping?
CLLM: [{"response": "Sorry, I cannot answer this."}]

Please think step by step and try to solve the problem. In the end output the answer.

User interaction:

User: $question
CLLM: [
```

Listing 4. Tool Template

```
You are assessing a submitted answer on a given task based on an ideal answer. Here is
 the data:
[BEGIN DATA]
***
[Task]: $input
***
[Submission]: $submission
***
[Ideal]: $ideal
***
```

```
[END DATA]
Do the submission and the ideal answer match?
First, write out in a step by step manner your reasoning about the criterion to be sure that your conclusion is correct.
Avoid simply stating the correct answers at the outset.
Then print only the single character "Y" or "N" (without quotes or punctuation) on its own line corresponding to the correct answer.
At the end, repeat just the letter again by itself on a new line.
If the question and answer are about ITE and revenue increase: The term ITE and revenue increase mean the same thing.
If the question asks about the best intervention or the best partner it is enough if the submission answers this correctly.

Reasoning:
```

Listing 5. Evaluation Template