EVALUATING CAUSAL REASONING IN LLMS: A CASE STUDY IN THE CHALLENGES OF PROCEDURAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) appear to represent and reason about a large range of phenomena. In particular, recent studies have examined whether LLMs can accurately reason about causation. One of the most commonly used benchmarks for evaluating formal causal reasoning in LLMs is CLADDER, a framework for procedural generation of benchmark data sets that was released in 2023. Experiments with CLADDER appear to indicate that LLMs can perform formal causal reasoning in a large percentage of generated examples. In this paper, we examine what can be reliably inferred when LLMs perform well on test sets produced by CLADDER. We identify several examples of superficial and highly predictive patterns that the CLADDER generator embeds in data sets it generates. We find that even simple models such as random forests perform well on CLADDER-generated data sets after training on a sample of generated data. We also find that the performance of many commercial LLMs on CLADDER test sets is extremely brittle and that removing large sections of the text that are critical for formal causal reasoning has almost no effect on performance. Given these results, the most likely hypothesis for the strong performance of state-of-the-art LLMs on CLADDER is that they are exploiting superficial patterns learned from exposure to training data from the generator. Any inference that LLMs are exhibiting more complex causal reasoning must rule out this hypothesis.

1 INTRODUCTION

One of the most important and challenging questions about large language models (LLMs) is the extent to which these models exhibit complex reasoning. On the one hand, LLMs regularly respond to prompts in ways that appear to require accurate reasoning about logic, mathematics, physics, and causation, as well as knowledge about the natural, physical, and social worlds (Hendrycks et al., 2021; Wei et al., 2022; Chen et al., 2021; Kiciman et al., 2024). On the other hand, much of this apparent knowledge and reasoning could be recalled verbatim from the massive amounts of data used to train LLMs, as well as some lightweight taxonomic knowledge gleaned from how humans produce natural language (Carlini et al., 2022; Collins et al., 2022). In short, researchers have sought to determine where LLMs sit between the extremes of artificial general intelligence and stochastic parrots (Zečević et al., 2023).

A major challenge in resolving this question has concerned evaluation — how to create large sets of questions and (correct) answers to evaluate specific types of reasoning. Early evaluations often repurposed existing questions originally designed to assess human knowledge and reasoning, including college entrance examinations, intelligence tests, and textbook exercises (e.g., Achiam et al., 2023; Hendrycks et al., 2021). However, because many such benchmarks were publicly available before contemporary LLMs were trained, researchers raised concerns that their content may have appeared in model training corpora and been memorized by LLMs (Xu et al., 2024; Sainz et al., 2023). These concerns are particularly difficult to resolve for closed models whose training data are not publicly documented (Deng et al., 2024). Subsequent analyses have found evidence of benchmark-specific overfitting and possible memorization in both open and closed models. (e.g., Zhang et al., 2024). As one researcher has quipped: "Pretraining on the test set is all you need" (Schaeffer, 2023).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

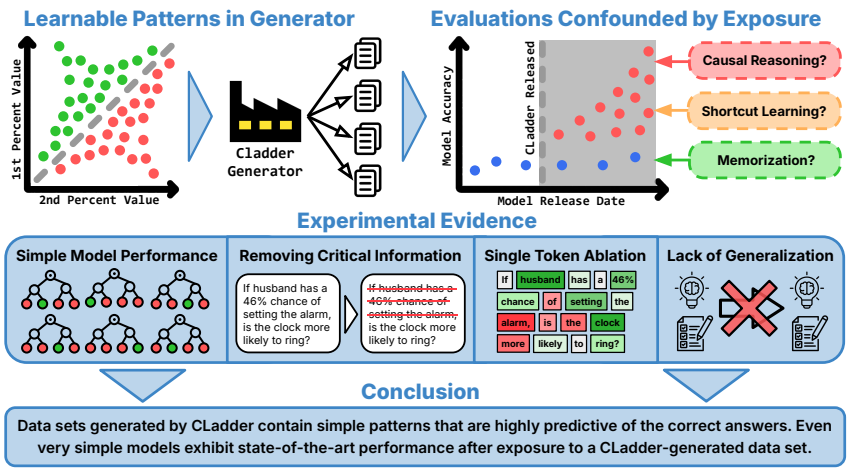

Figure 1: This paper examines how and why LLMs perform well on CLADDER, a dynamic benchmark for evaluating causal reasoning, and what can be inferred from the results.

Given the problems with static benchmark datasets, an alternative approach to evaluating LLM reasoning has gained prominence: *procedural generation*. With procedural generation, test sets are generated algorithmically rather than being generated manually or gathered from the real world (Chen et al., 2025; Risi and Togelius, 2020). In the context of evaluating LLM reasoning, procedural generation allows new test sets to be generated as needed, making it highly unlikely that test set instances could have been memorized (Zhu et al., 2024; White et al., 2025). Such "dynamic" test sets appear to overcome some of the most obvious problems associated with earlier "static" test sets. Procedural generation is best suited for reasoning tasks that have been sufficiently formalized such that questions can be randomly sampled from an extremely large (potentially infinite) space and their correct answers can be computed or verified automatically (Wei et al., 2025). Fortunately, many reasoning tasks have well-established formal theories that allow for the construction of procedural generators. These include mathematics, statistics, logic, and (most importantly for this paper) causal reasoning (e.g., Saxton et al., 2019; Wei et al., 2025; Jin et al., 2023).

In this work, we examine CLADDER, a particularly prominent and important procedural generator for evaluating the causal reasoning abilities of LLMs. Specifically, we focus on a single research question: ***What can be reliably inferred about the reasoning ability of an LLM when it performs well on a CLADDER-generated test set?*** We focus on this question because a large range of papers have implicitly or explicitly concluded that high performance on CLADDER-generated test sets implies that a given LLM has learned to perform formal causal inference, and that such reasoning capabilities will generalize to other scenarios. In contrast, our findings strongly indicate that such inferences are unwarranted. A far more likely conclusion is that high-performing models have been exposed to one or more CLADDER data sets and that the models have learned a set of superficial patterns that allow for high performance. While the ability to learn such patterns is impressive, it is not indicative of formal causal reasoning, and such performance is unlikely to generalize beyond data sets generated by CLADDER.

As summarized in Figure 1, we show that:

- **CLADDER embeds superficial predictive patterns in the data it produces** — We show that CLADDER's data generating process explicitly embeds superficial patterns in every data set that it produces. These patterns are simple and predictive of correct answers.
- **These patterns are learnable from exposure to a single CLADDER data set** — We show that a variety of simple and complex models can learn these superficial patterns and perform well on subsequent test sets. Simple models such as random forests can learn them when given superficial features derived from the queries. Open-weight LLMs can also learn these predictive patterns from Fine-tuning on CLADDER-generated queries and answers.

- **Exposure to CLADDER datasets is likely for many modern LLMs** — CLADDER-generated data has been popular and has proliferated widely since the paper's initial publication. CLADDER data is likely to be included in the training corpus of many LLMs.
- **Seemingly essential information is not required for high performance** — Some information in CLADDER queries is known to be essential for formal causal reasoning. We show that removing such information does not substantially impair LLM performance.
- **Seemingly incidental information is critical to model performance** — We show that omitting seemingly incidental tokens from specific CLADDER queries can dramatically affect the answers provided by LLMs.
- **High-performance on CLADDER does not generalize** — Models that achieve high accuracy on CLADDER perform no better than random on CounterBench, a recently released benchmark for causal reasoning with a very similar overall design and phrasing of causal queries.

2 CLADDER: BACKGROUND, DESIGN, AND PRIOR RESULTS

Causal reasoning is often identified as one of the most important and complex forms of reasoning that humans do, and thus causal reasoning has become an explicit target of efforts to evaluate the wider reasoning capabilities of LLMs. Several prominent researchers have explicitly identified causal reasoning as a key gap in the reasoning capabilities of LLMs (Zhang et al., 2023; Wu et al., 2024).

Early attempts to empirically evaluate causal reasoning in LLMs concentrated on commonsense causality, assessing whether models correctly assess everyday causal knowledge expressed in natural language (Roemmele et al., 2011). However, such evaluations conflate causal *knowledge* with causal *reasoning*, since a model could answer correctly by manipulating linguistic patterns rather than reasoning about the abstract causal concepts expressed in that language. Surveys of this work (e.g., Wu et al., 2024) note that while models can generate causal language that sounds accurate, they fall short of genuine causal reasoning.

The introduction of the CLADDER benchmark (Jin et al., 2023) was a substantial milestone in empirical evaluations of causal reasoning in LLMs. CLADDER leverages a formal theory of causal reasoning (Pearl, 2009) to create a procedural generator of question-answer pairs that range across ten distinct types of causal reasoning (average treatment effect, backdoor adjustment, collider bias, correlation, deterministic counterfactual, effect of treatment on the treated, explaining away, marginal effect, natural direct effect, and natural indirect effect) and that span Pearl's three rungs of causation (association, intervention, and counterfactual reasoning).

2.1 WHAT IS CLADDER?

CLADDER is a procedural generator of question-answer pairs. Questions are generated by randomly sampling from several sets of data structures that completely specify a causal reasoning task (e.g., problem type, graphical model, probability values, etc.) and then converting this information into a query in natural language. Correct answers to the query are also produced automatically by construction, based on the formal data structures. All queries have binary (yes/no) answers. Full details can be found in the original paper (Jin et al., 2023).

Such a procedural generation approach would appear to entirely escape the problems of prior "static" test sets that could have appeared in the training data of an LLM. Consequently, the results in the original paper (and subsequent papers that have used CLADDER) have far more credibility than prior work that used static test sets. It is worth noting that, in addition to the CLADDER generator, Jin et al. released several demonstration test sets that illustrate the ability of the generator to produce data with different properties (e.g., use of nonsense words in place of common nouns and causal dependencies that defied "commonsense" expectations). The full demonstration dataset contains 41,748 queries.

To avoid memorization issues, we generated a new dataset for our experiments with 9866 unique samples that includes all the combinations of options available using the standard implementation of the CLADDER generator. The dataset contains 1200 queries for each query type, unless there

are fewer than 1200 unique possibilities, in which case we include them all. See Appendix A.3 for further details on our generated dataset.

2.2 IMPACT OF CLADDER

The CLADDER paper had a strong effect on both the methods and the conclusions of the AI research community. In less than three years since its publication, the paper has been cited over 380 times, and CLADDER data has been used in more than 20 other empirical evaluations of LLM reasoning (e.g., Yu et al., 2025; He et al., 2026; Momentè et al., 2025; Mao and Venkat, 2026; Schrader et al., 2024). In some cases, those studies used the CLADDER generator. In others, however, the original demonstration test sets were used as static benchmarks.

The results in the original paper and subsequent papers that used CLADDER have been widely cited as evidence that LLMs can perform effective causal reasoning. It is worth noting that Jin et al. were quite cautious in the original paper, saying that "our work suggests that LLMs may still be far from reasoning reliably about causality." That said, they also claim that some LLM prompting methods presented in the paper "elicit sound causal reasoning in LLMs and help them solve challenging causality questions." Subsequent authors have made substantially stronger claims based on results from CLADDER in the original and subsequent papers (e.g., Gui and Ji, 2025; Chen et al., 2026a).

Only a few papers have attempted to understand the underlying source of the observed performance in ways that could clarify what can be inferred from high performance. Jin et al. themselves attempted to evaluate the degree to which surface linguistic information in queries affected the answers provided by LLMs. They substituted nonsense words for nouns in causal queries and observed essentially no change in performance. Similarly, Pacchiardi et al. analyzed a variety of LLM reasoning benchmarks for surface linguistic features that predict correct answers, and conclude that CLADDER test sets are among the least susceptible to learning of generator-specific shortcuts.

3 EXPERIMENTS AND KEY FINDINGS

In this section, we report the findings from analyses and experiments that inform our view on what can be inferred when a model accurately answers queries in CLADDER-generated data sets.

3.1 HIGHLY PREDICTIVE PATTERNS ARE DIRECTLY EMBEDDED BY THE GENERATOR

The methods used to generate CLADDER queries introduce multiple patterns into the data that can be learned and exploited by a model. The data generator produces each query by combining elements from different templates depending on the specified variables and causal graph. This method of generating queries leads to predictable elements that are repeated in any data produced by the generator, such as the structure and specific phrasing of the queries (see Appendix A.7 for additional details). In many cases, these patterns provide shortcuts that allow a model to correctly answer the queries without the need for any causal reasoning.

For example, after providing the graph structure, many of the average treatment effect queries have the form "$P(X = 1|Y = 0) = p_1, P(X = 1|Y = 1) = p_2$. Will $do(Y = 1)$ increase the chance of $X = 1$?" This query can be trivially solved by comparing p_1 and p_2 and checking the polarity of the query (i.e. whether it is phrased as "increase" or "decrease"). Once that pattern is apparent, other elements of the query, including the structure of the graph and names of the variables, are no longer necessary. In Figure 2, we plot the first probability given in the average treatment effect queries versus the second given probability. We can see that for each polarity the queries can be classified nearly perfectly using only the two probability values. This is not an isolated case: A very similar pattern can be observed in queries that ask about the effect of treatment on the treated. We note that some of the examples shown in Figure 2 have as many as seven probability values provided in the query, even though this highly predictive pattern uses only the first two.

The patterns above are fairly easy to identify, but we have identified other patterns in CLADDER-generated data that are far more subtle and difficult to detect by inspection alone. Without a way to mitigate this issue, there is a risk that any model exposed to the data during training may learn superficial shortcuts associated with the data generation procedure. This will likely lead to strong performance on the CLADDER datasets but poor generalization.

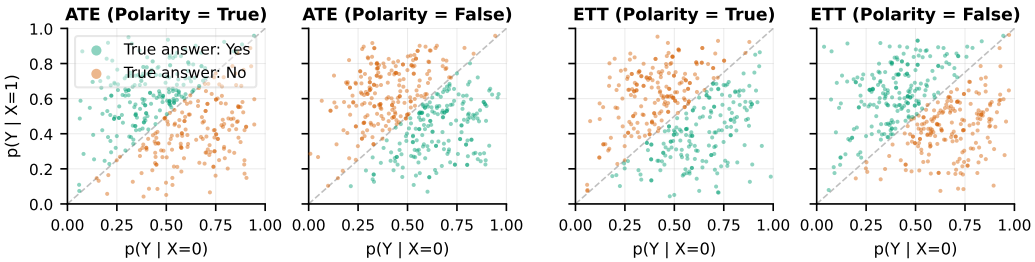

Figure 2: Scatter plots of the two key conditional probabilities provided in average treatment effect (ATE) and effect of treatment on the treated (ETT) queries, colored by correct answer. Within each polarity condition, yes and no answers are nearly linearly separable. A model that learns this simple comparison rule can achieve near-perfect accuracy on these query types without any formal causal reasoning.

3.2 Simple models can learn the predictive patterns

We have shown that predictive patterns exist within CLadder data, but can these patterns be learned by models and, if so, how well do they work? To answer these questions, we created two sets of superficial features derived directly from the query text: a small set of surface-level features that characterize the query text, as well as a fixed-length BERT embedding (Devlin et al., 2019) of each query. The BERT embeddings do not represent the probabilities that appear in the query, so they are appended to the embedding (see Appendix A.8 for feature descriptions).

We used each set of features to learn a random forest classifier that predicts the binary answer for each query. The cross-validated accuracy estimates for each feature set, disaggregated by query type, are shown in Figure 3. For some query types (e.g., collider bias and marginal), the random forest classifier was able to predict the answer with very high accuracy (100% and 94%, respectively). Averaged across all query types, the classifier achieved a cross-validated accuracy of 87%, roughly comparable to the accuracy achieved by current language models.

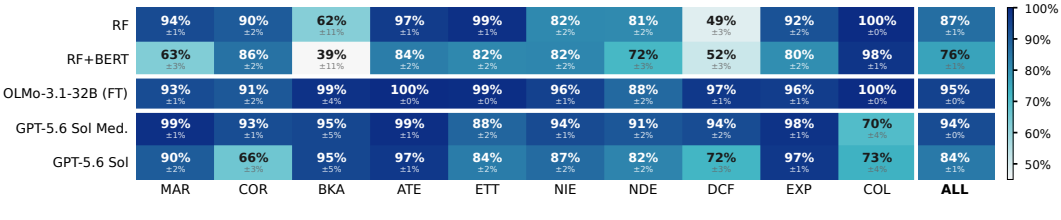

Figure 3: Model accuracies by query type. The first two rows show accuracy by query type on our generated CLadder test set for a random forest classifier trained on superficial features derived from the query text or BERT embeddings. We report the mean accuracy over 10-fold cross-validation. The third row shows the performance of OLMo-3.1-32B after fine-tuning on CLadder. We also show gpt-5.6-sol with and without reasoning for comparison.

To explore the effect of CLadder exposure on LLMs, we also investigate whether supervised fine-tuning on CLadder queries produces significant improvements in performance. We select four instruction-tuned models for which prior exposure to CLadder is unlikely based on their release dates or the availability of information about their training data: OLMo-3.1-32B-Instruct, OLMo-3-7B-Instruct, Llama-3.1-8B-Instruct, and Qwen2.5-3B-Instruct. Consistent with limited or no prior exposure, zero-shot performance on the generated queries (n = 9,866) is close to chance for all four models, with accuracies of 63.9 $\pm$ 0.9 %, 55.3 $\pm$ 1.0, 55.7 $\pm$ 1.0 and 52.1 $\pm$ 1.0, respectively.

When trained on 2,000 labeled examples, all fine-tuned models exceeded 80% overall accuracy. When trained on 16,000 examples, every model exceeded 94% overall accuracy. These results, shown in Appendix A.5, indicate that all models improve at approximately the same rate, which is not what we would expect from models of such different sizes (Qwen2.5-3B vs OLMo-3.1-32B).

The performance of all the models scales with the number of labeled training examples the model has been exposed to, which is consistent with learning superficial features.

Taken together, these results show that combinations of superficial features are sufficient to perform at state-of-the-art levels on CLADDER test sets. Even limited exposure to CLADDER data is sufficient for models to learn generator-specific shortcuts that transfer across CLADDER datasets. At the very least, these results should create a new, minimum baseline for surprising performance of LLM-based models.

3.3 MODELS MAY HAVE BEEN EXPOSED TO CLADDER DATA

The ability of relatively simple models to quickly acquire the knowledge necessary for high performance in CLADDER casts substantial doubt on the conclusion that evaluations using the benchmark indicate any degree of formal causal reasoning. This is particularly true if datasets from CLADDER have been included in the training data of LLMs. Can we rule out such exposure?

CLADDER has been publicly available since its original release. It has been copied, forked, and redistributed across a large and still-growing number of public repositories. We do not claim that any single evaluation, model, or copy of the dataset is the source of this exposure risk, since any one of these many copies is sufficient to contaminate a model's training data regardless of whether it was ever exposed to the canonical release. That alone is enough to complicate the interpretation of any post-release evaluation's results, because high performance cannot cleanly be attributed to improvements in model reasoning abilities when exposure to the data (in some form, through some channel) cannot be ruled out.

We evaluate the performance of several models of varying size and complexity on the dataset we produced using the CLADDER generator 2.1. Figure 4 shows the performance of each model versus the date of the model's public release. We observe a clear trend of performance improving over time, with early evaluations finding results slightly above random, and a steady increase in performance leading up to current "reasoning" models, which regularly achieve over 90% accuracy. But what is the cause of this improvement in performance?

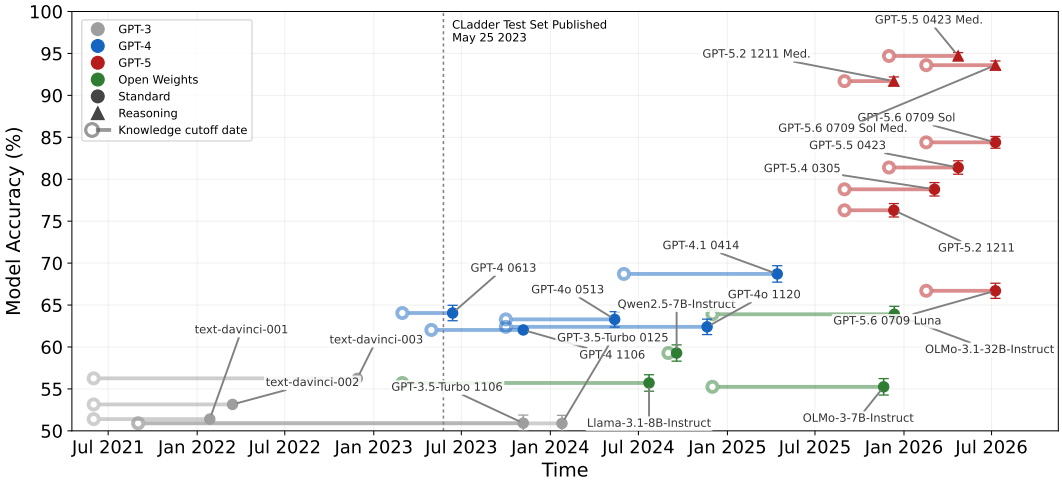

Figure 4: Model accuracy compared to the date of the model's release. Open circles represent the published "knowledge cutoff" dates for each model. The results for the text-davinci models come from the original CLADDER paper.

In general, model size and complexity has increased significantly over the time frame shown in Figure 4. However, many of these models were trained after the public release of the CLADDER generator and example datasets. Without access to the training corpus, there is no way to verify whether the models trained after the dataset's release were exposed during training. In light of our findings above, it appears likely that these models have learned the patterns embedded in the published datasets, confounding any evaluation of their causal reasoning abilities using CLADDER as a benchmark.

3.4 REMOVING ESSENTIAL INFORMATION HAS LITTLE EFFECT ON PERFORMANCE

Above we show that the use of simple features is sufficient to explain high performance on CLADDER. Next, we take a different approach: We examine whether LLMs still perform well on CLADDER queries when they lack the information that is necessary for formal causal reasoning. Specifically, we remove key components of the queries that leave them unsolvable by formal causal reasoning, and we find that such ablations produce no substantial drop in LLM performance.

Each query begins with a statement establishing a set of background conditions and specifying the structure of the causal graph used to generate the query (see Figure 5 for an example). Together, these elements are necessary to make the causal query "identifiable" (answerable with a unique quantitative solution). Our experiments remove each of these elements individually, and then together. In all cases, we observe minimal reductions in accuracy. Across five evaluated frontier models, these interventions resulted in reductions in overall accuracy of only 0.6 to 4.2 percentage points relative to the baseline performance on the unaltered queries.

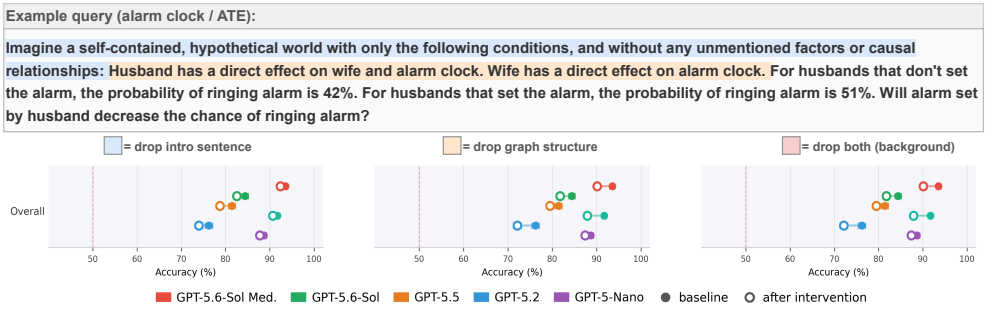

Figure 5: Model accuracy under interventions that remove elements of the queries that are critical for formal causal reasoning. A full query-wise breakdown in Appendix A.6. The 95% Wilson confidence intervals are not visible in this plot due to large sample size.

Next, we perform a set of even more aggressive interventions, resulting in queries where it is impossible to calculate the answer with the available information. We begin by removing only the first marginal or conditional probability value provided in the query, followed by removing only the second value provided. Finally, we remove all the marginal or conditional probability values that are provided with the query (identified as the "given_info" in the CLADDER queries). The result in all cases are queries with insufficient information to calculate the correct answer.

These interventions, shown in Figure 6, result in a larger reduction in accuracy than those in Figure 5, which makes sense due to the critical nature of the information being removed. However, we would expect performance to collapse to 50% in all three cases, as models should not be able to perform better than random chance. Instead, we see the accuracy reduced, but still remaining well above 50%. Across all three versions of the intervention, the models perform 11 to 21 percentage points above random.

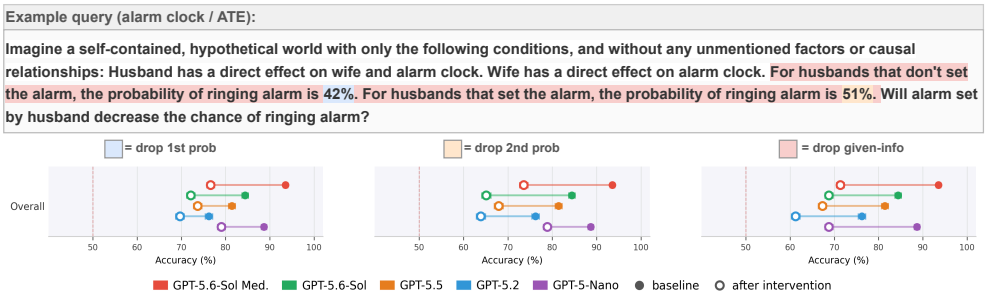

Figure 6: Model accuracy under interventions that remove elements of the queries that make it impossible to calculate the answer. Full query-wise breakdown in Appendix A.6. The 95% Wilson confidence intervals are not visible in this plot due to large sample size.

Taken together, these results indicate that the models are unlikely to be performing formal causal reasoning. A model that was doing so should not perform any better than random after such critical information is removed. In fact, these results show that large sections of the text that are critical for formal causal reasoning actually have a negligible effect on model performance. Instead, the models appear to be exploiting other shortcuts to correctly answer the queries, including ones that do not rely on elements of the query that formal causal reasoning methods would require.

3.5 SEEMINGLY INCIDENTAL INFORMATION GREATLY AFFECTS PERFORMANCE

The results above indicate that models largely ignore information that is essential for formal causal reasoning. That raises a natural question: What information *do* models use? In the next set of experiments, we aimed to localize the elements of the query that have the largest impact on model performance. We establish the model's baseline accuracy for each query, and then we re-process the query with a single token removed and observe any change in performance. The models we tested proved to be very brittle under these types of interventions, and highly sensitive to the presence or absence of specific tokens. We often see a large change in performance when tokens are removed that are incidental to a human's ability to solve the problem, and we find extreme variations in both baseline performance and sensitivity to individual tokens for nearly identical queries.

We find a surprising amount of variation in which tokens affect model performance, even between queries of the same type. Figure 7 shows the model's sensitivity to the removal of the token in each position for a set of queries drawn from the " marginal" queries. We run each query 100 times to establish a baseline accuracy, then 100 times for each token that is removed. We can see that despite being nearly identical in difficulty and appearance from a human's perspective, there is a significant variation in the response to the removal of each token. There are several tokens whose removal produces a complete collapse in accuracy ($\Delta = -1.0$), suggesting the model relies heavily on specific tokens. See Appendix A.10 for additional examples.



Figure 7: Token ablation results for a set of marginal queries on gpt-5.2. Each row corresponds to a single prompt, and each cell represents one token. Red cells denote tokens whose removal *reduces* model performance, while green cells denote tokens whose removal *increases* performance. Numerical tokens (#), percentage tokens (%), and the polarity indicator are always labeled. Otherwise, labels are shown only for tokens exceeding a salience threshold ($|\Delta| \geq 0.04$).

Looking closely at Figure 7 we can see the surprising variation we find in the model's response to nearly identical queries. For the first query, labeled $Q17820$, where the baseline accuracy is 100%, removing the single token "hard" in the compound word "hard-working" can reliably flip the model's answer such that its accuracy on the query is approximately 0%. The second query in the figure, labeled $Q7131$, is nearly identical aside from the variable names. Except in this case we observe a baseline accuracy of only 8.0%, even though these queries would be of equal difficulty for a human. Additionally, for the query with low baseline performance we find there are many tokens whose removal results in a significant increase in performance to nearly 100% accuracy. And again, we see that removing tokens that would appear inconsequential to a human seems to have a large impact on the model's performance.

Variation in the model's response to similar queries, as well as the surprising impact of individual tokens on model performance, are both strong indicators that the models are exploiting superficial patterns to solve the queries, and that disrupting these patterns can cause dramatic shifts in performance.

3.6 Model Performance on CLadder Does Not Generalize

The experiments described above establish that high performance on CLadder datasets is achievable through mechanisms that do not require formal causal reasoning. If models were performing formal causal reasoning, or even exploiting broadly generalizable patterns, we would expect these abilities to generalize to structurally similar causal reasoning tasks. If instead their performance is dependent on patterns that are specific to CLadder's data generation process, we would not expect those patterns to generalize to an independent benchmark.

We evaluate several frontier models on CounterBench (Chen et al., 2026b), a recently released benchmark for counterfactual causal reasoning that uses abstract variable names and was independently produced. Table 4 in Appendix A.9 shows the accuracy for each model on CounterBench. All frontier models score near chance on CounterBench (48.6% – 52.2%). The gap between CLadder and CounterBench performance for different models ranges from 29 to 44 percentage points.

Surprisingly, the random forest classifier trained exclusively on superficial features derived from query text transfers substantially better to CounterBench, achieving 79.1%, as shown in Table 4. It performed substantially above every frontier LLM we tested, suggesting that the frontier LLMs are not relying on the same transferable combination of features captured by the random forests or the fine-tuned models.

4 Discussion and Conclusion

The converging lines of evidence in Section 3 strongly suggest that high performance on CLadder does not reliably imply that LLMs are performing some type of formal causal reasoning. Instead, exposure to CLadder data and superficial pattern learning seems much more likely. Clearly, this has direct implications for what we should conclude from the results of prior evaluations that employed CLadder data sets and how the generator should be used in future studies.

More broadly, our results illustrate a limitation of procedural generation as a strategy for evaluating LLM reasoning. Many of the patterns in CLadder are not easily detected by inspection, and they appear to be an artifact of the generation process itself rather than resulting from design flaws that could be easily fixed. The combination of procedural generation and public examples of generated data may be incompatible with evaluating powerful machine learning models. While such an approach largely escapes the problem of a model memorizing training data that will also appear as test data, it does not escape the problem of a model learning a *generator's* intrinsic patterns.

This suggests at least three possible ways forward. First, in theory, a procedural generator could be constructed that is so faithful to the intended domain that its only embedded patterns are those that will generalize to real world systems. This seems unlikely, unless real-world systems are directly used to generate data. Second, a less faithful procedural generator could be constructed and then withheld from public view. Without public access to either the generator itself or data from the generator, its intrinsic patterns could not easily be learned. This strategy has already been taken in other evaluations of causal inference methods that have been somewhat successful (e.g., Dorie et al. (2019)). Third, if neither of these approaches is practical, then LLMs should always be compared to a baseline of a simpler model directly trained on the relevant public data set. This will, at least, prevent some types of unwarranted conclusions about the complex reasoning capabilities of LLMs.

AI use statement

We used LLMs during the preparation of this paper for code assistance. All experimental design, analysis, and interpretation of the results were done by the authors. No text in the paper was generated by AI.

Ethics statement

This work uses two existing public benchmark datasets (CLadder and CounterBench) and a newly generated dataset produced with CLadder's own procedural generator. No data involves human subjects, personally identifiable information, or sensitive content.

Reproducibility statement

We release our codebase which includes the intervention scripts, simple classifier code, and fine-tuning configurations as a zip-file during review and as a public GitHub link at camera-ready. Generated parameters and fine-tuning hyperparameters are documented in the appendix.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Edward Y Chang. 2026. Right for the wrong reasons: Epistemic regret minimization for causal rung collapse in llms. *arXiv preprint arXiv:2602.11675* (2026).

Junqi Chen, Sirui Chen, and Chaochao Lu. 2026a. Can Post-Training Transform LLMs into Causal Reasoners? *arXiv preprint arXiv:2602.06337* (2026).

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, et al. 2025. Benchmarking large language models under data contamination: A survey from static to dynamic evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 10091–10109.

Yuefei Chen, Vivek K Singh, Jing Ma, and Ruixiang Tang. 2026b. CounterBench: Evaluating and improving counterfactual reasoning in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 30350–30358.

Katherine M Collins, Catherine Wong, Jiahai Feng, Megan Wei, and Josh Tenenbaum. 2022. Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 44.

Nathaniël de Leeuw, Marceau Nahon, Mathis Reymond, Raja Chatila, and Mehdi Khamassi. 2025. Semantic Deception: When Reasoning Models Can't Compute an Addition. *arXiv preprint arXiv:2512.20812* (2025).

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 8706–8719.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314 [cs.LG] `https://arxiv.org/abs/2305.14314`

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, and Dan Cervone. 2019. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statist. Sci.* 34, 1 (2019), 43–68.

Jin Du, Li Chen, Xun Xian, An Luo, Fangqiao Tian, Ganghua Wang, Charles Doss, Xiaotong Shen, and Jie Ding. 2025. Ice Cream Doesn't Cause Drowning: Benchmarking LLMs Against Statistical Pitfalls in Causal Inference. *arXiv preprint arXiv:2505.13770* (2025).

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2, 11 (Nov. 2020), 665–673. https://doi.org/10.1038/s42256-020-00257-z

Longling Geng, Andy Ouyang, Theodore Wu, Daphne Barretto, Matthew John Hayes, Rachael Cooper, Yuqiao Zeng, Sameer Vijay, Gia Ancone, Ankit Rai, et al. 2026. Causalt5k: Diagnosing and informing refusal for trustworthy causal reasoning of skepticism, sycophancy, detection-correction, and rung collapse. *arXiv preprint arXiv:2602.08939* (2026).

Shurui Gui and Shuiwang Ji. 2025. Mitigating Spurious Correlations in LLMs via Causality-Aware Post-Training. arXiv:2506.09433 [cs.LG] https://arxiv.org/abs/2506.09433

Paul He, Yinya Huang, Mrinmaya Sachan, and Zhijing Jin. 2026. Uncovering Hidden Correctness in LLM Causal Reasoning via Symbolic Verification. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vera Demberg, Kentaro Inui, and Lluís Marquez (Eds.). Association for Computational Linguistics, Rabat, Morocco, 1231–1250. https://doi.org/10.18653/v1/2026.eacl-long.56

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. 2023. Cladder: Assessing causal reasoning in language models. *Advances in Neural Information Processing Systems* 36 (2023), 31038–31065.

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. 2024. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research* (2024).

Zihao Li and Fabrizio Russo. 2026. Leveraging Large Language Models for Causal Discovery: a Constraint-based, Argumentation-driven Approach. *arXiv preprint arXiv:2602.16481* (2026).

Jacqueline Maasch, John Kalantari, and Kia Khezeli. 2025b. CausalARC: Abstract Reasoning with Causal World Models. *arXiv preprint arXiv:2509.03636* (2025).

Jacqueline RMA Maasch, Alihan Hüyük, Xinnuo Xu, Aditya V Nori, and Javier Gonzalez. 2025a. Compositional causal reasoning evaluation in language models. *arXiv preprint arXiv:2503.04556* (2025).

Zhenjiang Mao and Anirudhh Venkat. 2026. Recurrent Confidence Chain: Temporal-Aware Uncertainty Quantification in Large Language Models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 18562–18566. https://doi.org/10.1109/ICASSP55912.2026.11461479

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics*. 3428–3448.

Filippo Momentè, Alessandro Suglia, Mario Giulianelli, Ambra Ferrari, Alexander Koller, Oliver Lemon, David Schlangen, Raquel Fernández, and Raffaella Bernardi. 2025. Triangulating LLM Progress through Benchmarks, Games, and Cognitive Tests. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 20051–20072. https://doi.org/10.18653/v1/2025.findings-emnlp.1092

Lorenzo Pacchiardi, Marko Tesic, Lucy G Cheke, and José Hernández-Orallo. 2024. Leaving the barn door open for Clever Hans: Simple features predict LLM benchmark answers. *arXiv preprint arXiv:2410.11672* (2024).

Judea Pearl. 2009. *Causality*. Cambridge University Press, Cambridge.

Sebastian Risi and Julian Togelius. 2020. Increasing generality in machine learning through procedural content generation. *Nature Machine Intelligence* 2, 8 (2020), 428–436.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning.. In *AAAI spring symposium: logical formalizations of commonsense reasoning*. 90–95.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 10776–10787.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing Mathematical Reasoning Abilities of Neural Models. In *International Conference on Learning Representations*.

Rylan Schaeffer. 2023. Pretraining on the test set is all you need. *arXiv preprint arXiv:2309.08632* (2023).

Timo Pierre Schrader, Lukas Lange, Simon Razniewski, and Annemarie Friedrich. 2024. QUITE: quantifying uncertainty in natural language text in Bayesian reasoning scenarios. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2634–2652.

Anjiang Wei, Yuheng Wu, Yingjia Wan, Tarun Suresh, Huanmi Tan, Zhanke Zhou, Sanmi Koyejo, Ke Wang, and Alex Aiken. 2025. SATBench: Benchmarking LLMs' Logical Reasoning via Automated Puzzle Generation from SAT Formulas. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 33832–33849. https://doi.org/10.18653/v1/2025.emnlp-main.1716

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, et al. 2025. Livebench: A challenging, contamination-limited llm benchmark. In *International Conference on Learning Representations*, Vol. 2025. 91595–91631.

Anpeng Wu, Kun Kuang, Minqin Zhu, Yingrong Wang, Yujia Zheng, Kairong Han, Baohong Li, Guangyi Chen, Fei Wu, and Kun Zhang. 2024. Causality for large language models. *arXiv preprint arXiv:2410.15319* (2024).

Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, et al. 2024. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244* (2024).

Kanta Yamaoka, Sumantrak Mukherjee, Thomas Gärtner, David Antony Selby, Stefan Konigorski, Eyke Hüllermeier, Viktor Bengs, and Sebastian Josef Vollmer. 2026. Linear-LLM-SCM: Benchmarking LLMs for Coefficient Elicitation in Linear-Gaussian Causal Models. *arXiv preprint arXiv:2602.10282* (2026).

Longxuan Yu, Delin Chen, Siheng Xiong, Qingyang Wu, Dawei Li, Zhikai Chen, Xiaoze Liu, and Liangming Pan. 2025. Causaleval: Towards better causal reasoning in language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 12512–12540.

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. 2023. Causal parrots: Large language models may talk causality but are not causal. *arXiv preprint arXiv:2308.13067* (2023).

Cheng Zhang, Stefan Bauer, Paul Bennett, Jiangfeng Gao, Wenbo Gong, Agrin Hilmkil, Joel Jennings, Chao Ma, Tom Minka, Nick Pawlowski, et al. 2023. Understanding causality with large language models: Feasibility and opportunities. *arXiv preprint arXiv:2304.05524* (2023).

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. 2024. A Careful Examination of Large Language Model Performance on Grade School Arithmetic. arXiv:2405.00332 [cs.CL] https://arxiv.org/abs/2405.00332

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Gong, Diyi Yang, and Xing Xie. 2024. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *International Conference on Learning Representations*, Vol. 2024. 18091–18128.

A Appendix

A.1 Critiques of CLadder and Synthetic Benchmarks

Several recent papers have identified limitations of CLadder from directions complementary to ours. Geng et al. (2026) and Chang (2026) argue that CLadder's template-heavy queries have low linguistic diversity and fail to capture failure modes that appear in naturalistic causal judgments, positioning their own benchmarks as testing more realistic reasoning scenarios. Maasch et al. (2025b) explicitly cite benchmark leakage as a motivation for designing CausalARC around randomly sampled tasks that cannot be memorized, noting that o4-mini offered unprompted CLadder-style outputs suggestive of training exposure. Maasch et al. (2025a) extend this concern with a compositional framework, finding that only o1 does not degrade with reasoning complexity, which is consistent with our findings that apparent causal competence collapses under targeted perturbations. Du et al. (2025) show that CLadder-style benchmarks do not test statistical pitfalls such as survivorship bias and collider conditioning in realistic scenarios, and find that model conclusions shift based on semantic label choice.

More recent work has examined specific subproblems: Li and Russo (2026) use LLMs for causal discovery via graph structure identification, while Yamaoka et al. (2026) benchmark coefficient elicitation in linear Gaussian Structural Causal Models. These complementary directions highlight that causal competence is multidimensional and that strong performance on one subproblem does not imply general causal reasoning ability.

A.2 Shortcut Learning and Benchmark Contamination

The phenomenon of models achieving high benchmark accuracy through spurious correlations rather than the intended reasoning process is well documented (Geirhos et al., 2020; McCoy et al., 2019). Chang (2026) formalize this as Rung Collapse — the systematic substitution of associational reasoning for interventional queries. de Leeuw et al. (2025) demonstrate analogous surface exploitation in arithmetic, showing that LLMs default to learned semantic associations over abstract reasoning when tested on novel symbol systems. On the contamination side, Zhang et al. (2024) show that models exhibit inflated performance on publicly available math benchmarks relative to held-out equivalents, with accuracy gaps correlating with the probability of generating training examples. Chen et al. (2026a) show that fine-tuning on synthetic SCM data produces dramatic accuracy gains similar to our LoRA experiments, but do not investigate whether those gains reflect genuine reasoning or shortcut learning.

A.3 Generated dataset

We generated a CLadder dataset using the original codebase of Jin et al. (2023), running `fig generate` across the configurations (`final/common` and `final/det-common`).

This produced 9,866 unique (question, answer) pairs, which we used for all of our experiments. We use this generated split rather than the original v1 release for our evaluation as our test set for the fine-tuning experiment.

Query-types vary in count because the underlying causal graph templates vary in richness and would generate duplicates. The original CLadder releases included surface-level interventions as a robustness test. We did not apply these during our dataset generation, as we found no significant difference in model performance between these dataset split.

To confirm our generated data matches the difficulty of the original CLadder v1 release, we evaluate gpt-5.5 (medium reasoning) zero-shot on our generated set and each of the four v1 splits. Table 2 shows nearly identical performance on all splits (all $p > 0.05$, none significant after Bonferroni correction).

Query type	Count	Rung
Marginal	1200	1
Correlation	1200	1
Collider bias	432	1
Backdoor adjustment	74	2
ATE	1200	2
ETT	1200	2
Exp-away	1200	2
NIE	1200	3
NDE	1200	3
Det-counterfactual	960	3
Total	**9,866**	

Table 1: Query-type distribution in the generated dataset.

Table 2: Zero-shot GPT-5.5 (medium reasoning) accuracy on our generated dataset (GENERATED) and each CLADDER v1 split individually. No split is significantly harder than the generated dataset (p-values from two-proportion z-tests, Bonferroni-corrected threshold $p < 0.0125$, none significant).

Split	N	Acc. (%)	95% CI	p-value
Generated	9,866	94.73	±0.4	—
Easy	10,559	94.16	±0.4	0.074
Hard	10,556	94.49	±0.4	0.443
Non-commonsense	10,240	94.11	±0.5	0.056
Anti-commonsense	10,392	95.20	±0.4	0.127

A.4 MODEL PERFORMANCE ON GENERATED CLADDER DATA

Table 3: Model accuracy on generated CLADDER queries (%, ± Wilson 95 % CI half-width). Bold = best per column. † Back-Adj has n=74; all other types $n \geq 432$.

Model	Overall	ATE	MAR	COR	EXP	NDE	DCF	NIE	ETT	COL	BKA
Non-LLM baselines											
RF	86.8±0.7	96.8±1.0	93.8±1.4	90.4±1.7	92.2±1.5	81.2±2.2	48.6±3.2	81.5±2.2	99.0±0.6	100.0±0.4	62.2±10.8
RF+BERT	76.2±0.8	83.8±2.1	62.7±2.7	85.8±2.0	80.2±2.3	71.5±2.6	51.7±3.2	81.8±2.2	81.6±2.2	97.9±1.4	39.2±10.9
Open-weight models											
Llama-3.1-8B-Instruct	55.7±1.0	79.5±2.3	52.2±2.8	46.3±2.8	52.8±2.8	51.2±2.8	61.0±3.1	49.0±2.8	60.8±2.8	39.8±4.6	50.0±11.1
Qwen2.5-3B-Instruct	52.1±1.0	87.6±1.9	47.8±2.8	54.0±2.8	15.4±2.0	55.4±2.8	54.5±3.1	54.0±2.8	49.8±2.8	50.0±4.7	50.0±11.1
OLMo-3-7B-Instruct	55.3±1.0	88.3±1.8	50.1±2.8	61.2±2.8	24.0±2.4	54.0±2.8	54.6±3.1	54.0±2.8	51.7±2.8	66.0±4.5	56.8±11.0
OLMo-3.1-32B-Instruct	63.9±0.9	97.3±0.9	54.5±2.8	53.2±2.8	70.1±2.6	70.7±2.6	63.2±3.0	60.2±2.8	47.5±2.8	50.0±4.7	51.4±11.1
GPT models											
GPT-3.5-turbo	55.5±1.0	74.9±2.5	49.4±2.8	53.0±2.8	40.4±2.8	55.3±2.8	52.1±3.2	55.4±2.8	63.9±2.7	55.6±4.7	40.5±10.9
GPT-4-0613	71.0±0.9	98.5±0.7	62.9±2.7	63.2±2.7	72.8±2.5	69.5±2.6	67.1±3.0	63.9±2.7	74.3±2.5	62.5±4.5	37.8±10.8
GPT-4o-2024-11-20	70.6±0.9	95.6±1.2	69.5±2.6	57.2±2.8	76.4±2.4	70.2±2.6	61.7±3.1	70.8±2.6	71.1±2.6	51.6±4.7	35.1±10.6
GPT-4.1-2025-04-14	75.6±0.8	94.4±1.3	74.2±2.5	73.0±2.5	87.4±1.9	72.0±2.5	69.9±2.9	71.3±2.6	69.3±2.6	56.9±4.6	62.2±10.8
GPT-5.2	76.3±0.8	94.2±1.3	74.9±2.5	66.6±2.7	92.3±1.5	70.5±2.6	62.2±3.1	72.9±2.5	82.4±2.2	50.7±4.7	81.1±8.8
GPT-5.2 (med. reasoning)	91.7±0.5	**99.6±0.4**	**99.1±0.6**	94.1±1.3	**98.4±0.7**	86.8±1.9	88.9±2.0	89.6±1.7	**88.6±1.8**	58.6±4.6	93.2±6.0
GPT-5.4-2026-03-05	78.8±0.8	92.8±1.5	83.3±2.1	77.5±2.4	93.5±1.4	72.7±2.5	64.2±3.0	68.8±2.6	82.5±2.1	56.0±4.7	82.4±8.6
GPT-5.5-2026-04-23	81.4±0.8	96.2±1.1	82.8±2.1	70.5±2.6	97.4±0.9	80.2±2.3	70.1±2.9	79.0±2.3	81.6±2.2	56.9±4.6	82.4±8.6
GPT-5.5 (med. reasoning)	**94.7±0.4**	**99.6±0.3**	98.9±0.6	**94.2±1.3**	98.3±0.7	89.6±1.7	**94.0±1.5**	93.3±1.4	**88.6±1.8**	**94.5±2.1**	92.4±5.5
GPT-5.6-luna	66.7±0.9	93.2±1.4	66.2±2.7	63.7±2.7	49.3±2.8	65.4±2.7	61.3±3.1	67.5±2.6	73.1±2.5	50.0±4.7	43.2±11.0
GPT-5.6-sol	84.4±0.7	97.4±0.9	90.2±1.7	65.6±2.7	97.3±0.9	82.2±2.2	72.2±2.8	87.2±1.9	84.1±2.1	73.1±4.2	**94.6±5.5**
GPT-5.6-sol (med. reasoning)	93.5±0.5	99.1±0.5	98.8±0.6	93.2±1.4	98.3±0.7	**91.1±1.6**	93.4±1.6	**93.5±1.4**	88.3±1.8	69.7±4.3	92.4±5.5

A.5 FINE-TUNING DETAILS

We fine-tune four open-weight models (OLMo-3-7B-Instruct, OLMo-3.1-32B-Instruct, Llama-3.1-8B-Instruct, and Qwen2.5-3B) on the public CLADDER dataset, stratified by query type at 80/20 split for train and val. We use the generated CLADDER dataset as our test set. We perform parameter-

efficient fine-tuning via LoRA (Hu et al., 2021). To study the effects of training data size, we vary the number of training examples $N \in 50, 200, 500, 1\text{k}, 2\text{k}, 4\text{k}, 8\text{k}, 16\text{k}$ and evaluate on the held-out test set. We use rank $r = 16$, scaling factor $\alpha = 32$, dropout 0.05, targeting both attention projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and feed-forward layers (`gate_proj`, `up_proj`, `down_proj`), following Dettmers et al. (2023). Models are trained for 5 epochs with AdamW, cosine learning rate 2×10^{-4}, effective batch size 16, bf16 precision with gradient checkpointing, and prompt tokens masked to -100 so only the answer token contributes to training loss.

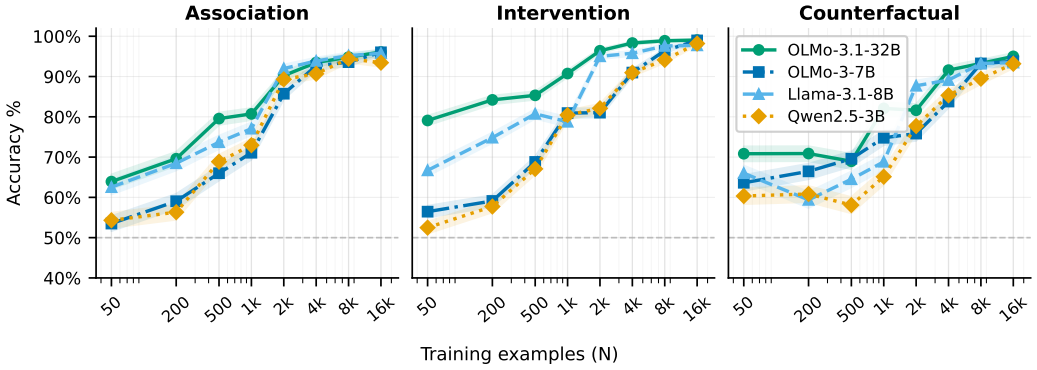

Figure 8: Model performance as a function of the number of publicly-available CLADDER samples the model is exposed to during fine-tuning.

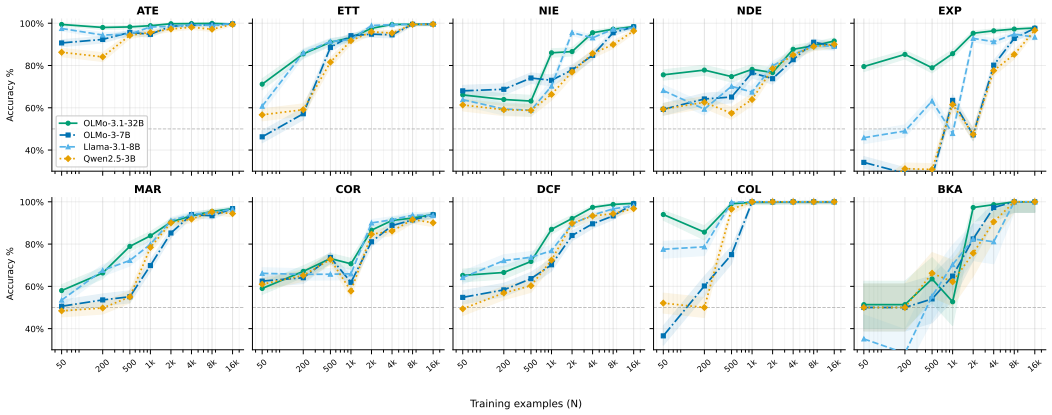

Figure 9: Model performance (per query-type) as a function of the number of publicly-available CLADDER samples the model is exposed to during fine-tuning.

A.6 QUERY-WISE ACCURACY AFTER PERFORMING INTERVENTIONS

We abbreviate query types present in CLADDER as follows: Marginal (MAR), Correlation (COR), Back-door Adjustment (BKA), Average Treatment Effect (ATE), Effect of Treatment on the Treated (ETT), Natural Indirect Effect (NIE), Natural Direct Effect (NDE), Deterministic Counterfactual (DCF), Explaining Away (EXP), and Collider Bias (COL). Figures 10 and 11 show the per-query-type accuracy change for the six interventions that were discussed in Section 3.4.

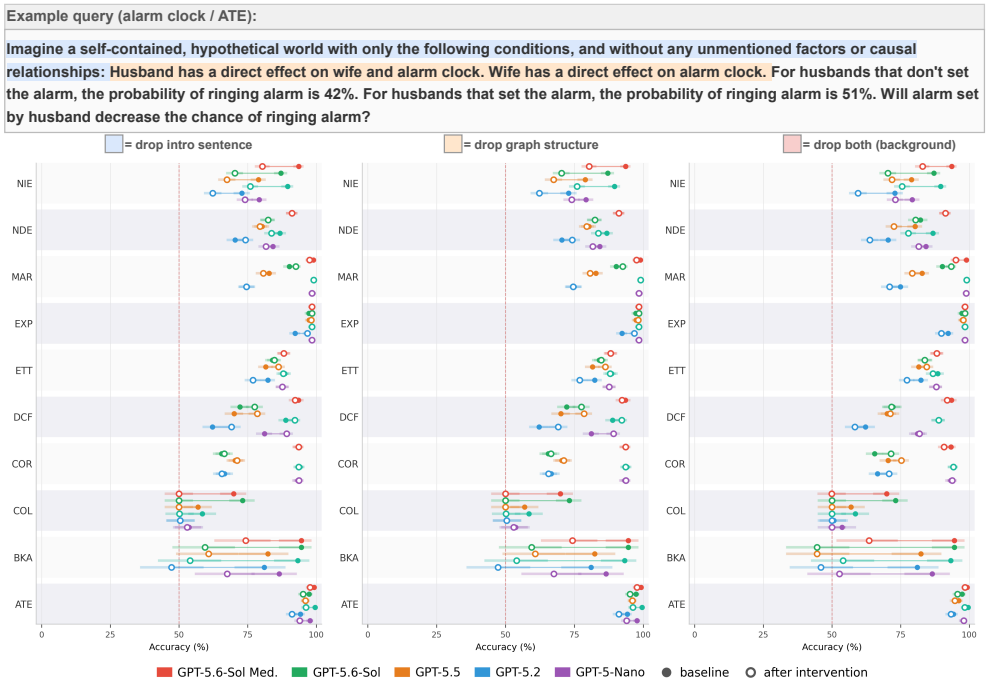

Figure 10: Per-query-type model accuracy under interventions that remove elements of the queries that are critical for formal causal reasoning. We show 95% Wilson CI.



Figure 11: Per-query-type model accuracy under interventions that remove elements of the queries that make it impossible to calculate the answer. We show 95% Wilson CI.

A.7 SOURCES OF DETERMINISM IN CLADDER DATA

To understand the mechanisms underlying model performance on CLADDER, we conducted a systematic analysis of the dataset's generation codebase. This analysis reveals three distinct levels at which answer determinism is introduced into the benchmark, each of which provides a potential shortcut a model could exploit without performing genuine causal reasoning.

A.7.1 PARAMETRIC DETERMINISM

CLADDER's `DifficultyBuilder` generates questions by sampling Structural Causal Model parameters from predefined specification lists stored in per-story configuration files. In the easy split, these specifications assign consistent sign directions to causal mechanisms across all 37 non-nonsense stories, implying the direction of the causal effect can be typically determined by reading a single mechanism edge in the prompt, without computing anything. The hard split systematically reverses these signs. In both cases, knowing which story a query comes from is often sufficient to predict the correct answer regardless of the specific probability values provided.

A.7.2 STRUCTURAL DETERMINISM

The collision graph topology ($X \rightarrow V \leftarrow Y$), Y has no parents and is therefore statistically independent of X by construction. Any intervention on X cannot affect Y, so the average treatment effect $E[Y|do(X = 1)] - E[Y|do(X = 0)] = 0$ for every parameter setting. All queries pairing a collision graph with an ATE, ETT or COR query type therefore have a predetermined answer that can be read from the graph structure.

A.7.3 FORMULA-LEVEL DETERMINISM

The COL query type computes the correct answer as $(-1)^{polarity}$, where polarity is a flag that controls whether the question is phrased as "does X affect Y" (answer is always *no*) or "does X not affect Y" (answer is always *yes*). This is independent of the causal graph, the Structural Causal Model parameters, and the numerical correlation value shown in the prompt. The correlation value is computed and displayed to the model, but is never used to determine the ground truth answer.

A.8 RANDOM FOREST CLASSIFIER FEATURES

When training the random forest classifier we use the following features:

- Boolean variable indicating whether the query asks if one quantity is "greater than" or "less than" another quantity.
- The number of characters in the query.
- The number of probability values that appear in the query.
- The first probability that appears in the query. This feature is set to 0 for queries with no probability values.
- The second probability that appears in the query. This feature is set to 0 for queries with less than two probability values.
- The last probability that appears in the query. This feature is set to 0 for queries with no probability values. Some query types contain 2 or less probability values, so a probability value that appears first or second in the query could also be the last.
- Boolean variable indicating whether the first probability value in the query is larger than the second.
- Boolean variable indicating whether the first probability value in the query is larger than the last.
- The product of the first and second probabilities that appear in the query.
- The product of the first and last probabilities that appear in the query.
- The product of the second and last probabilities that appear in the query.

- Boolean variables indicating the presence or absence of the following strings: ”does it mean,” ”does not effect,” ”method,”, ”case by case,” ”chance of,” ”increase,” ”decrease,” ”overall,” ”more likely,” ”less likely,” ”when observing,” ”larger,” ”smaller,” ”does the chance of,” ”does the probability of,” ”does the likelihood of,” ”disregard the mediation effect,” ”positively affect,” ”negatively affect,” ”through,” ”would it be,” ”instead of,” ”not,” ”and,” ”or,” ”doesn’t,” and ”correlation.”
- Boolean variables indicating if the query starts with the words ”for” or ”would.”

A.9 MODEL PERFORMANCE ON COUNTERBENCH

Model	**Type**	**Acc. (%)**
Simple models ($n = 1200$)		
Random Forest	Classifier	79.1 ± 2.3
Commercial models (one-word prompt, $n = 1200$)		
GPT-3.5-turbo	Base	48.6 ± 2.8
GPT-4.1	Base	48.8 ± 2.8
GPT-4o	Base	52.2 ± 2.8
GPT-5-nano	Base	51.2 ± 2.8
GPT-5.2	Base	50.8 ± 2.8
GPT-5.5	Base	51.9 ± 2.8
GPT-5.5	Med. reason	50.3 ± 2.8
GPT-5.6 Sol	Base	49.3 ± 2.8
GPT-5.6 Sol	Med. reason	52.1 ± 2.8
Open-weight base models ($n = 1200$)		
OLMo-3-7B-Instruct	Base	50.4 ± 2.8
OLMo-3.1-32B-Instruct	Base	50.4 ± 2.8
LLama-3.1-8B-Instruct	Base	50.8 ± 2.8
Qwen2.5-3B-Instruct	Base	50.5 ± 2.8
Fine-tuned models, $N = 4000$ ($n = 1200$)		
OLMo-3-7B-Instruct	Fine-tuned	63.0 ± 2.7
OLMo-3.1-32B-Instruct	Fine-tuned	71.5 ± 2.5
LLama-3.1-8B-Instruct	Fine-tuned	68.9 ± 2.6
Qwen2.5-3B-Instruct	Fine-tuned	59.2 ± 2.8

Table 4: CounterBench Alpha accuracy $\pm$ half-width of Wilson 95% CI (chance = 50%). All commercial GPT models perform at or near chance regardless of generation. Fine-tuned open-weight models and the random forest outperform all commercial models.

A.10 Expanded Token Ablation Experiment

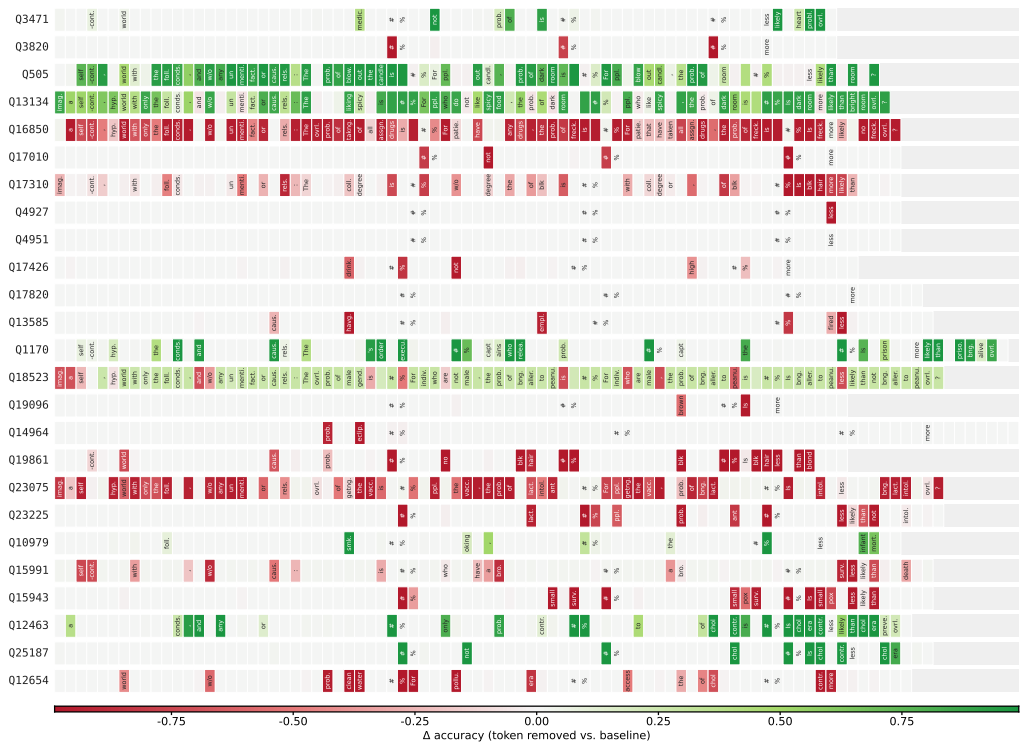

Figure 12: Token ablation analysis on gpt-5.2 for a random sample of 25 evaluated marginal queries. Same visualization as Figure 7. Color encoding, label conventions, and salience threshold are identical to the main figure.