arXiv:2505.18931v2 [cs.AI] 19 Sep 2025

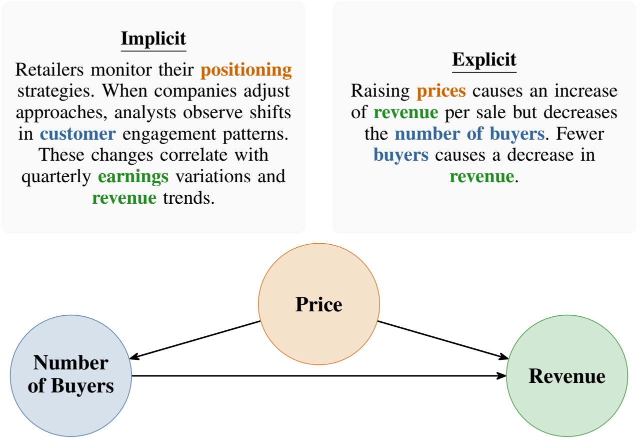
Can Large Language Models Infer Causal Relationships from Real-World Text?

Ryan Saklad[1], **Aman Chadha**[2,**], **Oleg Pavlov**[1], **Raha Moraffah**[1]

[1]Worcester Polytechnic Institute
[2]Amazon Gen AI

Abstract

Understanding and inferring causal relationships from texts is a core aspect of human cognition and is essential for advancing large language models (LLMs) towards artificial general intelligence. Existing work evaluating LLM causal reasoning primarily focuses on synthetically generated texts which involve straightforward causal relationships that are explicitly mentioned in the text. This fails to reflect the complexities of real-world tasks. In this paper, we investigate whether LLMs are capable of inferring causal relationships from real-world texts. We develop a benchmark drawn from real-world academic literature which includes diverse texts with respect to length, complexity of relationships (different levels of explicitness, number of nodes, and causal relationships), and domains and sub-domains. To the best of our knowledge, our benchmark is the first-ever real-world dataset for this task. Our experiments on this dataset show that LLMs face significant challenges in inferring causal relationships from real-world text, with the best-performing model achieving an average F1 score of only 0.477. Through systematic analysis across aspects of real-world text (degree of confounding, size of graph, length of text, domain), our benchmark offers targeted insights for further research into advancing LLM causal reasoning.



Implicit

Retailers monitor their **positioning** strategies. When companies adjust approaches, analysts observe shifts in **customer** engagement patterns. These changes correlate with quarterly **earnings** variations and **revenue** trends.

Explicit

Raising **prices** causes an increase of **revenue** per sale but decreases the **number of buyers**. Fewer **buyers** causes a decrease in **revenue**.

Figure 1: An example causal graph illustrating the difference between explicit and implicit texts describing the same causal relationships. The explicit text directly states causal relationships using clear language. The implicit text describes the same relationships through indirect language requiring inference (e.g., "monitor their positioning strategies" and "observe shifts in engagement patterns"). This highlights a key challenge in causal discovery from real-world texts, where reasoning must be used to accurately construct the graph.

Introduction

The ability to identify and understand causal relationships embedded within texts is a fundamental aspect of human intelligence (Pearl 2009; Harari 2014) and is crucial for complex decision-making. Humans excel at inferring these relationships when given a text, even when they are not explicitly stated; for instance, from "These changes correlate with quarterly earnings valuations and earnings trends" humans must first infer that there is a relationship between the nodes described earlier in the passage and the one currently described. Then, they must reason that both mentions of earnings are intended to be a single concept. Finally, they must reason to determine the direction of causality. Large language models (LLMs) have shown remarkable progress, fueling aspirations towards artificial general intelligence (AGI). Given the importance of causality to intelligence, significant research has focused on assessing the causal reasoning capabilities of LLMs. Much of this work evaluates LLMs' stored causal knowledge about nodes (Zhou et al. 2024) rather than their ability to infer relationships from textual contexts. While some studies examine extracting causal relationships from text (Veldhuis et al. 2024; Hosseinichimeh et al. 2024; Oh 2025; Jin et al. 2024; Joshi, Ahmad, and Modi 2024; Lasheras and Pinheiro 2025), they often use synthetically generated or simplified texts with explicitly stated causal links. This approach, however, falls short of real-world scenarios where causal relationships are often embedded in long, complex texts, with varying degrees of explicitness, complex causal structures, and diverse domains. Figure 1 illustrates this contrast: while synthetic texts may present clear, explicit causal statements, real-world texts often convey causal relationships far more implicitly, requiring moving beyond mere comprehension to

[**] Work done outside position at Amazon Gen AI.

extract the underlying causal structure.

To address this gap, we introduce Realistic Causal Structure from Text (**ReCAST**), a novel benchmark drawn from real-world academic literature. ReCAST features diverse texts varying in length, complexity of relationships (including different levels of explicitness, number of nodes, and causal relationships), and spanning multiple domains and sub-domains, mirroring the challenges of real-world causal reasoning from text. Our experiments on state-of-the-art LLMs using ReCAST demonstrate significant struggles, with the best models achieving an average F1 score of only 0.477. Our results reveal that LLMs particularly struggle with implicitly stated causal information, with performance degrading substantially as causal relationships become less explicit in the text. By systematically characterizing these deficiencies across multiple dimensions, our benchmark offers targeted insights for further research into advancing LLM causal reasoning.

Our main contributions are as follows:

- We propose ReCAST, the first benchmark to evaluate real-world causal reasoning abilities of LLMs from text, featuring 292 samples drawn from academic literature with diverse text lengths (684-171,141 characters), graph complexities (5-140 nodes, 6-205 edges), and varying degrees of causal explicitness across multiple domains.
- We conduct extensive experiments on five state-of-the-art LLMs, revealing significant limitations in LLM's causal reasoning from text, with the best-performing model achieving only 0.477 F1 score.
- We develop a novel LLM-as-a-Judge evaluation framework that enables automated, fine-grained assessment of causal graphs across multiple dimensions (node/edge precision/recall, semantic similarity, abstraction levels), and provide systematic analysis showing that model struggles persist even when node identification is simplified, indicating causal reasoning as the primary bottleneck.

Related Work

Causal Reasoning with LLMs. As LLMs have grown in capabilities, significant research has explored their causal reasoning abilities. Existing benchmarks often assess these capabilities through various lenses. Some focus on identifying causal structures from pre-defined node name lists or datasets (as surveyed in (Kıcıman et al. 2024) or parts of CausalBench (Zhou et al. 2024)), largely bypassing complex textual inference. Other benchmarks utilize textual inputs, but frequently rely on synthetic or simplified narratives with explicit, often pairwise, relationships or for constructing relatively small graphs (e.g., CLadder (Jin et al. 2024), ExpliCa (Miliani et al. 2025), From text to map (Hosseinichimeh et al. 2024)). Other prior work evaluates causal understanding via question-answering, probing LLMs' stored knowledge and their ability to deduce specific causal claims from short prompts or contexts (e.g., CaLQuest.PT (Lasheras and Pinheiro 2025), COLD (Joshi, Ahmad, and Modi 2024)). While some studies engage with real-world texts, they target more constrained tasks such as identifying a single primary causal link (Oh 2025) or classifying individual sentences for causality (Veldhuis et al. 2024), rather than requiring the synthesis of a complete causal graph. These approaches, detailed further in the Appendix, generally do not address the challenge of causal graph construction under the full diversity and complexities of real-world conditions, where causality is often implicit and embedded within long narratives. ReCAST differs from prior work by directly targeting this gap, using academic literature as verified source data for causal reasoning from text under realistic conditions.

Knowledge Graph Discovery with LLMs Recent years have witnessed substantial progress in knowledge graph construction frameworks with LLM-based approaches (Bratanic 2024; Zhang and Soh 2024; Yu et al. 2023). These methods convert unstructured text into structured formats by identifying entities, relationships, and additional attributes. However, they emphasize open-domain and factual information, differing from the unique challenges of causal reasoning.

The ReCAST Dataset

In this section, we present the construction of ReCAST, a realistic causal discovery from text benchmark. Each benchmark sample requires a source text, a ground-truth causal graph, and adherence to our realism criteria. We achieve this through a systematic 3-step pipeline: (i) Paper collection and filtering, (ii) Annotation, (iii) Post-processing. We elaborate on each step below.

Paper Collection and Filtering

We focus ReCAST on the economics domain due to the unique characteristics of some of its causal literature, which is especially well-suited for this benchmark. These papers are ideal for a text-based causal benchmark, as they contain detailed textual descriptions of their goals, methodology, background, and causal reasoning, while avoiding reliance on non-textual elements (e.g., numerical data). Because each paper centers around a single, human-authored causal graph, adheres to a standardized structure, and articulates modeling assumptions, causal graphs can be accurately annotated, and construction from the text is feasible, even when elements are not mentioned explicitly. At the same time, economics is highly cross-disciplinary, spanning many domains, which offers substantial diversity while allowing for standardized quality control. Statistics detailing the distribution of academic domains within ReCAST are presented in the Appendix.

To support large-scale collection, we utilize API access to economics papers. To ensure our corpus allows for legal redistribution of texts and diagrams, we restrict our search to the open access repositories (MDPI and PLOS). We use a keyword search for the term "causal loop diagram", a commonly used term in economics literature to describe causal graphs, resulting in 646 candidate papers. We exclude workshop papers by removing papers whose abstracts include the keywords "workshop" or "group model build", as these texts do not include sufficient details about the graph to make it identifiable. We then manually exclude all papers with graphs unsuitable for annotation, including ones with no or multiple unrelated causal graphs, purely illustrative or non-causal diagrams, and causal graphs with poor legibility or ambiguous graph elements. We retain only papers where there is a single causal graph - either as the sole causal graph, or as the culmination of a sequence of graphs - so that it can serve as the ground-truth causal graph for benchmarking. During the manual filtering, the location of each primary causal graph in the paper is labeled to ensure accurate annotation (e.g., “Top diagram on page 7”). A more comprehensive comparison can be found in the Appendix.

Annotation

As ReCAST is a text-based benchmark, it is important that ground-truth causal graphs are converted from images to a text-based representation to be used as the ground-truth answer. We find through manual evaluation that vision-based LLMs are unsuitable for annotation due to frequent hallucinations. Therefore, we employ domain experts to annotate each causal graph into a standardized format (“source_variable” -¿ “sink_variable”). They are provided detailed instructions for the annotation process, including which graph elements to include, a standardized output format, and step-by-step examples. To avoid potential errors, we further instruct annotators to flag any papers that cannot be unambiguously annotated given the instructions, which we manually review for suitability.

Post-Processing

Graph Post-Processing To achieve ReCAST’s goal of being a realistic causal reasoning benchmark, it is essential that the ground-truth answers are accurate. While human annotators are highly accurate at transcribing causal graphs to our standardized textual format, they are prone to minute errors. To mitigate the risk of erroneous ground-truth answers, we utilize a rigorous post-processing pipeline for ground-truth graphs. We first utilize code-based approaches to identify formatting mistakes, and attempt string matching approaches to automatically correct them. Any samples that are unable to be corrected for formatting are flagged and have the formatting manually corrected.

To verify transcription quality, a second annotator relabeled 37 randomly chosen diagrams (778 edges, 616 nodes; $\approx$ 15% of ReCAST). We observed 22 missing and 5 spurious edges, giving edge–level precision $=$ 0.995, recall $=$ 0.977, F_1 $=$ 0.986, and Cohen’s κ $=$ 0.94. Node labels showed 8 auto-correctable typos (e.g., `carbondioxide`) and 4 minor prefix/suffix omissions; no major name mismatches occurred. Full per-graph statistics and the Python script used to compute them are provided in the Appendix. We explore the possibility of utilizing code-based approaches for correction of node naming, but find that it is prone to erroneously combining distinct nodes (e.g., GDP and GNP). Therefore, we utilize LLMs for automated correction, which is detailed further in the Appendix.

Text Post-Processing With the causal graphs finalized, it is important to ensure that the suitability of source texts. As ReCAST is a text-based benchmark, the PDF must first be converted to a textual format. As markdown is a common format for LLMs, we choose this as our target for conversion. We use the Python library PyMuPDF to extract the raw text for each paper. The output of this step contains numerous formatting errors, such as arbitrary line breaks. We utilize a multi-step LLM pipeline, as code-based approaches are infeasible given the diversity of documents, and manual testing shows current LLMs cannot perform this process accurately in one step.

Attribute	**Mean ± SD**	**Range**
Number of Samples	–	292
Text Length	40541 ± 17722	(684, 171141)
Nodes per Graph	25.0 ± 15.8	(5, 140)
Edges per Graph	37.4 ± 24.3	(6, 205)
Confounding per Node	0.123	–

Table 1: ReCAST dataset statistics showing its real-world attributes across key dimensions. Text length is measured in number of characters. The wide ranges in text length, graph complexity, and node explicitness reflect the natural diversity found in real-world academic literature, making this a challenging and realistic benchmark for causal reasoning evaluation.

We first prompt Mistral Small (Team 2025a) to convert from the PDF text to well-structured markdown. The goal of this step is to remove non-textual elements (which are impossible to accurately represent in markdown) and entirely irrelevant elements (to streamline the task and save on computational costs). Therefore, Mistral is tasked to output the markdown auto-regressively while skipping over non-textual elements (such as images, charts, or other figures), in-line citations, references, publication information, and appendices. The output of this step is a well-formatted markdown version of the paper.

Normalization Tool

```
{
 "normalizations": [
   {"start": "text to find (start)",
    "end": "text to find (end)",
    "replacement": "new text"},
    ... ]
}
```

However, this may include information that makes the task trivial (such as a table including all of the sample’s causal relations), or it may erroneously reference removed elements (resulting in an internally inconsistent document). To correct these issues, we utilize o3-mini (OpenAI 2025) to remove explicit references to the causal graph and correct any references to missing elements using a normalization tool. Explicit references to the graph are unrealistic (such as a table listing every connection in the graph), as the task is to create a causal graph that does not already exist. However, we are careful to not remove other information about

Model	Node Precision (↑)	Node Recall (↑)	Edge Precision (↑)	Edge Recall (↑)	F1 (↑)	SHD (↓)	Normalized SHD (↓)
QwQ	0.881 ± 0.119	0.488 ± 0.216	0.802 ± 0.201	0.242 ± 0.200	0.450 ± 0.193	$\mathbf{36.860 \pm 21.793}$	0.107 ± 0.084
Llama-8B	0.827 ± 0.191	0.359 ± 0.244	0.677 ± 0.265	0.125 ± 0.160	0.302 ± 0.201	41.197 ± 23.631	0.120 ± 0.102
Qwen-32B	0.862 ± 0.117	0.434 ± 0.242	0.747 ± 0.231	0.181 ± 0.170	0.381 ± 0.200	40.602 ± 23.886	0.112 ± 0.089
o3-mini	0.862 ± 0.134	0.459 ± 0.231	0.806 ± 0.197	0.208 ± 0.185	0.415 ± 0.200	38.481 ± 22.120	0.107 ± 0.090
R1	$\mathbf{0.893 \pm 0.108}$	$\mathbf{0.522 \pm 0.217}$	$\mathbf{0.817 \pm 0.183}$	$\mathbf{0.260 \pm 0.196}$	$\mathbf{0.477 \pm 0.187}$	38.193 ± 22.491	$\mathbf{0.105 \pm 0.097}$

Table 2: Comparison of different models' performance on ReCAST (mean ± standard deviation).

the graph to avoid making it unidentifiable. To make these changes, we provide the model a normalization tool to minimize output tokens, and unnecessary changes, while allowing for code-based validation.

Experiments

In this section, we examine the performance of state-of-the-art LLMs on ReCAST. Specifically, we investigate the following research questions: (i) How well do current LLMs perform on real-world causal reasoning from text? (ii) How does performance vary across different characteristics of ReCAST, including text length, degree of confounding, number of nodes, and number of causal relationships? Finally, we demonstrate common LLM reasoning failures via a case study.

Experiment Setup

Models. We assess 5 representative instruction-tuned and reasoning-focused LLMs against ReCAST, including Deepseek-R1 (et al. 2025), o3-mini (OpenAI 2025), QwQ (Team 2025b), Qwen-32B (Team 2024), and Llama-3-Instruct-8B (Grattafiori et al. 2024) with size of open-source models from 8B to 671B. These models exhibit proficiency in following instructions, long-context inputs, and generating appropriately formatted responses.

We assess in the zero-shot setting, using default hyperparameters, and excluding samples where the maximum context length is exceeded. For proprietary models, we retry responses where we fail to receive an answer. To isolate causal discovery abilities from determining the intended level of abstraction of the graph, the expected number of nodes is provided to the LLM. We enforce strict JSON formatting for graph outputs to enable automated evaluation. For any response that fails formatting requirements, we apply a post-processing step using Mistral Small (Team 2025a) to convert malformed outputs into valid JSON while preserving the intended relationships. For judge outputs, we use a standardized YAML format, with answers automatically extracted using code, and invalid responses automatically rejected and retried. The complete rubric and scoring mechanics are detailed in the Appendix.

Evaluation Metrics. In order to fairly evaluate ReCAST's graph construction from text task, it is important to assess each included node and edge, and check for omission of each node or edge in the ground-truth graph. Traditional causal discovery metrics like Structural Hamming Distance (SHD), precision, recall, and F_1 are infeasible for this task, requiring exact matching of node names, and cannot determine if a constructed relation is supported by the text when omitted from the ground-truth.

To address these limitations, we design a LLM-as-a-Judge for evaluation, which can assess semantic similarity, abstraction levels, text-grounding, and other factors. Our LLM-as-a-Judge framework evaluates causal graphs across four key dimensions: node-level precision, node-level recall, edge-level precision, and edge-level recall. For precision evaluation, each predicted node or edge is compared against both the ground-truth graph and the source text along three aspects:

- **Presence**: whether the concept appears in the reference (strong match, weak match, or no match)
- **Semantic similarity**: how closely the meaning aligns (strong, moderate, weak, or not applicable)
- **Abstraction level**: whether the concept is broader, aligned, or narrower than the reference

For recall evaluation, each ground-truth element is also assessed for its *importance* (core, intermediate, or peripheral) and how well it is captured in the predicted graph. The judge processes each evaluation separately to maintain focus and accuracy. For precision, if an element receives "no match" against both the ground-truth graph and source text, it scores 0.0; otherwise, the higher of the two composite scores is used. For recall, we compute a weighted average where correctness scores are multiplied by importance weights, ensuring that accurately capturing more critical causal relationships contributes more significantly to the final score. Using this, we can calculate traditional metrics for performance, as detailed below.

Performance on Real-World Causal Reasoning

Table 2 presents the performance of assessed models on ReCAST. All LLMs perform poorly, with the best-performing model, R1, achieving just an average F_1 of 0.477 across all samples. This shows how even state-of-the-art LLMs struggle at causal reasoning from real-world text. Reasoning models showed the best performance, and there was a positive trend between model size and performance. Comparing across metrics, models perform in roughly the same ranking, showing broad agreement for overall performance. Raw SHD scores are high, as expected, due to the large size of graphs. However, normalized SHD is low, due to most graphs being sparse. Larger models and reasoning models appear to do notably better on this task. Notably, all models exhibit significantly lower recall than precision, showing that models have an easier time generating nodes and edges that are valid from the source text, but not the same as in the ground-truth graph. Due to the realistic nature of the benchmark, there is a high standard deviation across metrics, with difficulty varying based on factors including degree of confounding, number of nodes, number of edges, text size, and domains mimicking the diverse conditions for real-world causal reasoning.

Model	Precision (↑)	Recall (↑)	F$_1$ (↑)	SHD (↓)
R1	**0.513**	**0.512**	**0.502**	40.6
QwQ	0.485	0.502	0.483	**40.3**
o3-mini	0.455	0.491	0.459	40.7
Qwen-32B	0.290	0.435	0.332	51.0
Llama-8B	0.219	0.409	0.267	58.6

Table 3: Name-assisted performance where models are provided ground-truth node names. Performance remains poor despite removing node identification uncertainty, highlighting that causal reasoning is the fundamental bottleneck.

Node Identification vs. Causal Reasoning To investigate the extent to which poor model performance is explained by node identification or deeper causal reasoning, we conduct an ablation experiment in which models are explicitly provided with the complete set of ground-truth node names. Under these conditions, evaluation is deterministic, isolating the effects of LLM-based grading and node identification from that of edge causality.

Table 3 shows results from this controlled scenario. While models trivially achieved perfect node-level precision and recall by design, the expected improvements in causal inference were limited: edge-level F$_1$ slightly improved for stronger models such as `R1` (by only $+0.025$) and `o3-mini` (by $+0.044$). Meanwhile, weaker models like `Qwen-32B` and `Llama-8B` saw reductions in performance, declining by -0.049 and -0.035 respectively. SHD similarly showed minor improvements, suggesting that explicitly provided node schemas do not have a consistent effect on performance.

Though deterministic and LLM-as-a-Judge evaluation is not directly comparable, both approaches show similar results. Despite removing entity recognition uncertainty, models continue to perform poorly at correctly inferring causal relationships, with causal reasoning as a fundamental bottleneck. This ablation highlights that the poor performance of LLMs on ReCAST is due to fundamental limitations in their causal reasoning capabilities and the effectiveness of our LLM-as-a-Judge evaluation method.

Performance Analysis

We conduct a performance analysis to provide in-depth insights into ReCAST. Specifically, we investigate which factors of real-world text affect LLM performance. Specifically, we study: degree of confounding, text length, number of nodes, number of edges, and domain.

Degree of Confounding We investigate whether poor overall performance can be partially attributed to variations in sample difficulty. To do so, we analyze samples using the degree of confounding for its ground-truth graph, which measures how much information a source text contains about its ground-truth graph. In order to systematically measure this, we develop a confounding score that categorizes each node into three levels of explicitness: (1) *explicit* - the node name or a clear synonym appears directly in the text, (2) *implicit* - the node concept is mentioned indirectly or can be reasonably inferred, and (3) *absent* - the node does not appear in the text whatsoever. We use `R1` to automatically label each node in every sample using a detailed rubric, then calculate the degree of confounding as:

$$\mathrm{DC} = \frac{1}{|V|} \sum_{v \in V} \begin{cases} 1, & \text{if } v \notin E, \\ 0, & \text{if } v \in E \end{cases}$$

where E represents explicitly mentioned nodes and V is the set of all nodes. This acts as a natural measure of difficulty, as explicitly mentioned nodes are easier to identify than those requiring inference or entirely absent from the text. The confounding score provides a quantitative way to assess how much causal reasoning (versus simple text comprehension) is required for each sample.

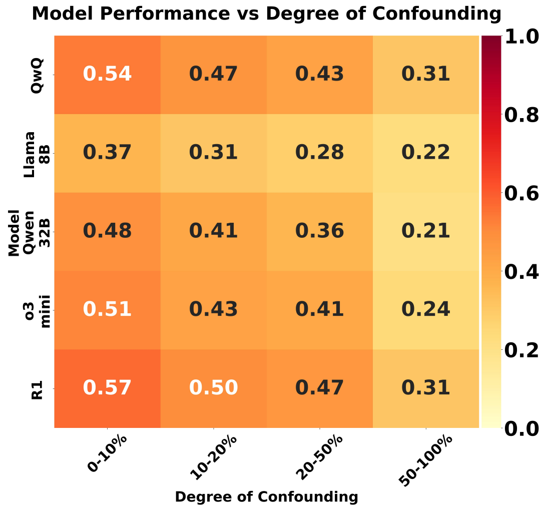


Figure 2: Heatmap of the average model scores across confounding bins (lenient definition), showcasing how degree of confounding has a large impact on model performance. It underscores LLMs' struggles to infer causal relationships when explicit references are sparse.

Samples with high degrees of confounding are rare in a realistic benchmark. Therefore, we divide samples into four sections to ensure that each is sufficiently large to show the effect of confounding. As shown in Figure 2, there is a strong positive correlation between a sample's degree of confounding and model performance (reported using F$_1$). Additionally, all models struggle even for the samples with the lowest degree of confounding, showing that LLMs struggle with the task even under easier conditions. This performance further degrades for the highest degree of confounding, with `F1` dropping by around half for all models between the samples with lowest and highest degrees of confounding. We analyze this relationship in further detail using an alternate, stricter definition of confounding in the Appendix.

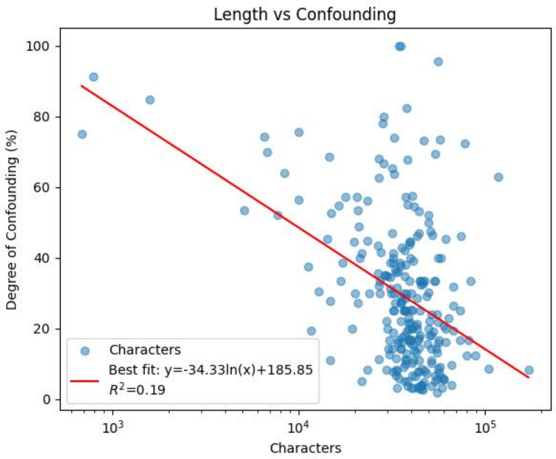


Figure 3: Relationship between text length and degree of confounding. This scatter plot shows each sample's character count on the x-axis versus its degree of confounding on the y-axis. There is a modest negative correlation (R^2 = 0.187), indicating that longer texts tend to include fewer unobserved confounders. This helps explain why models perform slightly better on larger input instances, as reduced confounding makes causal edges easier to identify.

Effect of Size As ReCAST is a realistic benchmark, samples are diverse with respect to size of source text and ground-truth graph. We organize samples into quartiles for length of source text, number of edges, and number of nodes. We report the difference in performance between the small samples and large samples in Table 4. Counterintuitively, there is a weak, but positive correlation between size and performance. Given the strong effect of Degree of Confounding on performance shown in Section , we investigate the relationship between degree of confounding and size. As samples greatly vary in size, we analyze each of them using a log scale. We find that text length, number of nodes, and number of edges each have a small, negative correlation with degree of confounding (with R^2 values of 0.187, 0.002, and 0.006 respectively). We attribute some of the performance gains to this, which is further explored in the Appendix.

Effect of Domain To assess the domain coverage of ReCAST, we create 6 domain categories, aiming to minimize overlap between domains while ensuring that all samples belong to at least one domain. The domain categories are: Agriculture & Food Systems, Economics & Public Policy, Education, Engineering & Technology, Environmental & Earth Sciences, and Medicine. We use `R1` to classify each sample by domain. We manually review each sample not classified as Economics & Public Policy for suitability in ReCAST, excluding those that were correctly classified as non-economics. As each ReCAST benchmark sample includes the economics domain, we analyze by excluding this domain from samples.

Model	Text	Edges	Nodes
QwQ	+0.02 ± 0.28	+0.00 ± 0.27	+0.06 ± 0.27
Llama-8B	–0.01 ± 0.28	+0.01 ± 0.28	–0.000 ± 0.27
Qwen-32B	+0.06 ± 0.29	–0.13 ± 0.29	–0.09 ± 0.29
o3-mini	+0.03 ± 0.29	–0.06 ± 0.30	+0.06 ± 0.28
R1	+0.04 ± 0.25	+0.00 ± 0.29	+0.02 ± 0.25

Table 4: Difference in model performance (mean ± std) between the top and bottom quartiles of text length, edge count, and node count. Positive values indicate improved performance on larger instances.

On average, each sample spans 2.35 domains, with Engineering & Technology and Environmental & Earth Sciences being the most prevalent. Model performance did not largely differ across samples with different numbers of domains (e.g., 1 domain vs 4), indicating that interdisciplinary samples do not explain poor overall benchmark performance. This is explored further in the Appendix.

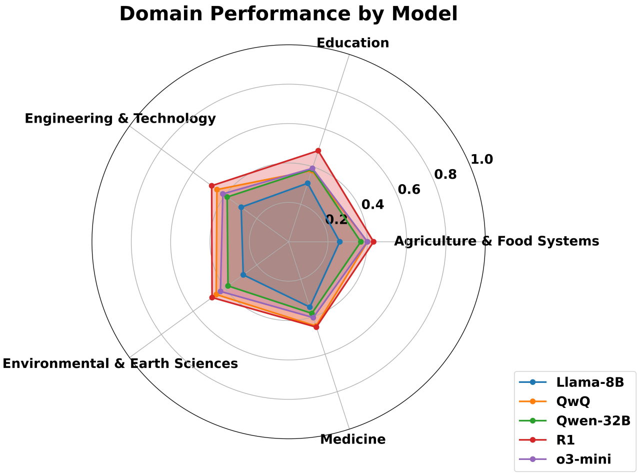


Figure 4: Radar chart of domain-specific accuracies for each model, depicting the accuracy of different models across specific domains. Variability in performance across domains highlights the contextual challenges faced by models in domain-specific causal inference, pointing towards the need for domain-sensitive model adjustments.

As shown in Figure 4, domain has a small effect on performance. The best-performing model overall, `R1`, similarly performs the best on all domains, except for Agriculture & Food Systems, where `QwQ` slightly outperforms. However, all models perform poorly overall regardless of domain, showing that this is not a major contributor to overall performance. Variation in performance across domains indicates that ReCAST captures the variations in causal reasoning performance between models under realistic conditions.

Additional Experiments

Effect of Knowledge Cutoff To investigate whether performance on ReCAST is influenced by pre-training con-

R1 Chain-of-Thought Excerpt

> ...In the Materials and Methods, they mention... auxiliary variables (savings, non-farm income, per capita arable land), and constants (region area, **precipitation**, etc.)...
> County's arable land area (18) -> Cash crop cultivation (10) (if less land, shift to cash crops)... Cash crop cultivation (10) -> Farmers' income (maybe part of non-farm income, but non-farm income is separate. Cash crop would be part of farm income... county financial capital increases lead to infrastructure and loan access...
> - Loan usage flexibility -> Diversified investments (cash crops, etc.)

Figure 5: Verbatim excerpts from reasoning trace. Ellipses added for readability and bold for emphasis.

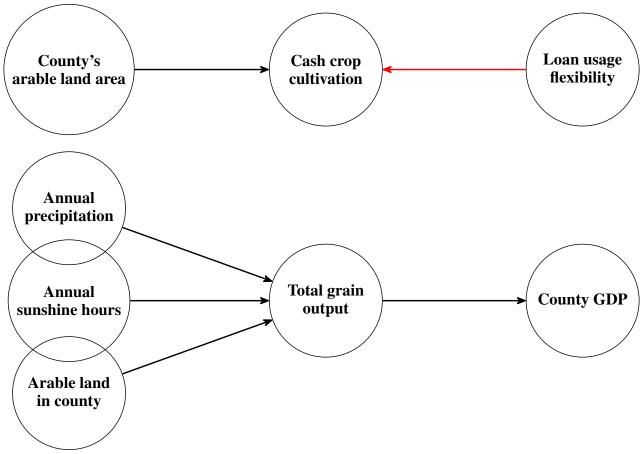


Figure 6: Top: `R1`-generated model subgraph, with spurious links labeled in red. Bottom: the ground-truth causal subgraph.

tamination, we conduct an analysis comparing performance for each model before and after their knowledge cutoff. To ensure sufficient sample size, we use `Llama-8B` and `o3-mini`, with knowledge cutoffs are prior to 2024. Our dataset contains 35 samples derived from papers published in 2024 or later. For `Llama-8B`, the average F1 score increased from 0.300 on older papers to 0.315 on recent papers, while for `o3-mini`, the average F1 score increased from 0.410 to 0.452. Despite this variation in performance, it is not statistically significant, with Mann-Whitney U test p-values of 0.5356 and 0.4074 respectively. This suggests that models' ability to identify causal relationships is not related to training on these specific documents during pre-training.

Case Study To further analyze ReCAST, we select a benchmark sample about livelihood efficiency in the Qinba Mountains and the graph constructed by the best-performing model, `R1`. For illustrative purposes, we focus on a subset of the graph showing the causal relationship between various climate and land factors with grain production and overall output. The source text describes these relationships most clearly in this passage:

> Land and climate are the fundamental conditions for agricultural production. ... therefore, sunshine, precipitation, and arable land area were selected to represent the natural capital of the county.

While this relationship is explicitly described, `R1` fails to faithfully include this in its graph. As shown in Figure 5, it initially identifies precipitation as a relevant factor, but ultimately overlooks inclusion of every climate driver in its final causal graph. Despite their explicit mentions, *Annual precipitation* and *Annual sunshine hours* fail to be included. Similarly, `R1` identifies how cash crop cultivation is a part of farm income, which is similar to GDP, but fails to ultimately include this. It also swaps the broader node *Total grain output* for a narrow focus on *Cash crop cultivation*. Further, it generates an erroneous link between loan usage flexibility and cash crop cultivation; however, loan flexibility relates to financial capital and coordination mechanisms, not crop selection decisions. We show the relevant ground-truth edges for this subgraph in Figure 6. `R1` only partially captures one relationship by oversimplifying, omits three entirely, and adds another erroneously. This shows how, even with straightforward textual cues, evaluated LLMs still struggle at causal reasoning. We include a complementary human expert case study for this sample in the Appendix.

Conclusion

Understanding causal relationships from text, a core aspect of human cognition, is essential for advancing LLMs towards artificial general intelligence. However, evaluating this skill under real-world conditions has been limited by the lack of appropriate benchmarks. This paper introduced ReCAST, which is, to our knowledge, the first benchmark to assess LLM causal reasoning capabilities from text under realistic conditions. ReCAST draws diverse samples from academic literature, featuring texts varied in length, relational complexity, and domain, thereby incorporating the challenges of real-world causal reasoning.

Extensive experiments utilizing ReCAST revealed that state-of-the-art LLMs struggle significantly with this task, with the best-performing model achieving an average F1 score of just 0.477. In-depth analysis identified common pitfalls, including challenges with implicitly stated information and distinguishing relevant causal factors from surrounding contextual details across lengthy passages. Notably, performance remained poor even under low degrees of confounding or when node identification was simplified, indicating the primary bottleneck lies in causal reasoning itself. ReCAST offers a robust platform for future research and benchmarking investigating these deficiencies, and is cost-effective, with the cost of all experiments for all LLMs, including ablations and judging, remaining under $250. It serves as a valuable tool for the research community, enabling more targeted investigations into LLM limitations. These insights directly inform the development of next-generation models, advancing efforts towards more sophisticated causal reasoning abilities in LLMs.

References

Bratanic, T. 2024. Building knowledge graphs with LLM graph transformer: A deep dive into LangChain's implementation of graph construction with LLMs. Towards Data Science.

et al., D.-A. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv preprint arXiv:2501.12948.

Grattafiori, A.; Dubey, A.; Jauhri, A.; et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Harari, Y. N. 2014. *Sapiens: A Brief History of Humankind*. New York: Random House.

Hosseinichimeh, N.; Majumdar, A.; Williams, R.; and Ghaffarzadegan, N. 2024. From text to map: a system dynamics bot for constructing causal loop diagrams. *System Dynamics Review*, 40(3): e1782.

Jin, Z.; Chen, Y.; Leeb, F.; Gresele, L.; Kamal, O.; Lyu, Z.; Blin, K.; Adauto, F. G.; Kleiman-Weiner, M.; Sachan, M.; and Schölkopf, B. 2024. CLadder: Assessing Causal Reasoning in Language Models. https://arxiv.org/abs/2312.04350.

Joshi, A.; Ahmad, A.; and Modi, A. 2024. COLD: Causal reasOning in cLosed Daily activities. arXiv:2411.19500.

Kıcıman, E.; Ness, R.; Sharma, A.; and Tan, C. 2024. Causal Reasoning and Large Language Models: Opening a New Frontier for Causality. arXiv:2305.00050.

Lasheras, U. A.; and Pinheiro, V. 2025. CaLQuest.PT: Towards the Collection and Evaluation of Natural Causal Ladder Questions in Portuguese for AI Agents. In Hettiarachchi, H.; Ranasinghe, T.; Rayson, P.; Mitkov, R.; Gaber, M.; Premasiri, D.; Tan, F. A.; and Uyangodage, L., eds., *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, 325–343. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Miliani, M.; Auriemma, S.; Bondielli, A.; Chersoni, E.; Passaro, L.; Sucameli, I.; and Lenci, A. 2025. ExpliCa: Evaluating Explicit Causal Reasoning in Large Language Models. arXiv:2502.15487.

Oh, S. S. 2025. Plausibly Exogenous Galore. Notion. Available at https://sangmino.notion.site/1a897b8106ca44eeaf31dcd5ae5a61b1?v=ff7dc75862c6427eb4243e91836e077e.

OpenAI. 2025. OpenAI o3-mini System Card. OpenAI, January 31, 2025. https://cdn.openai.com/o3-mini-system-card.pdf.

Pearl, J. 2009. *Causality*. Cambridge: Cambridge University Press.

Team, M. A. 2025a. Mistral Small 3.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models!

Team, Q. 2025b. QwQ-32B: Embracing the Power of Reinforcement Learning.

Veldhuis, G. A.; Blok, D.; de Boer, M. H. T.; Kalkman, G. J.; Bakker, R. M.; and van Waas, R. P. M. 2024. From text to model: Leveraging natural language processing for system dynamics model development. System Dynamics Review, 40(3): e1780.

Yu, S.; Huang, T.; Liu, M.; and Wang, Z. 2023. BEAR: Revolutionizing Service Domain Knowledge Graph Construction with LLM. In Monti, F., Rinderle-Ma, S., Ruiz Cortés, A., Zheng, Z., and Mecella, M. (Eds.), Service-Oriented Computing. ICSOC 2023. Lecture Notes in Computer Science, vol 14419, pp. 339–346. Springer, Cham.

Zhang, B.; and Soh, H. 2024. Extract, Define, Canonicalize: An LLM-based Framework for Knowledge Graph Construction. arXiv preprint arXiv:2404.03868.

Zhou, Y.; Wu, X.; Huang, B.; Wu, J.; Feng, L.; and Tan, K. C. 2024. CausalBench: A Comprehensive Benchmark for Causal Learning Capability of LLMs. arXiv:2404.06349.

Technical Appendix

Ryan Saklad[1], **Aman Chadha**[2,**], **Oleg Pavlov**[1], **Raha Moraffah**[1]

[1]Worcester Polytechnic Institute
[2]Amazon Gen AI

Detailed Comparison with Causal Reasoning Benchmarks

This section provides a comparative overview of ReCAST against existing benchmarks for causal reasoning. While each benchmark contributes to understanding LLM capabilities, ReCAST is specifically designed to test the construction of complex causal graphs from long-form, real-world academic texts. We evaluate benchmarks based on their diversity and complexity. In this context, we consider a benchmark diverse when it spans many domains or sub-domains, or draws data from many different types of sources. Meanwhile, we deem a sample complex when there are varied degrees of confounding in samples; that is, that the benchmark features many samples that are not highly explicit, requiring that causal reasoning, rather than mere reading comprehension, to be used. Lastly, we also examine the realism (of the source data) - is it drawn from the real-world? The following table offers a high-level visual comparison, with further details on each benchmark discussed subsequently.

Discussions of Selected Benchmarks

The following discussions provide context for the data presented in Table 1, highlighting their approaches and how we notably differ from this prior work. While these works are valuable, prior work fails to measure the causal reasoning abilities of LLMs from text under real-world settings. We highlight the approach of each benchmark, and how they compare to ReCAST.

Several benchmarks concentrate on pairwise causal relations, include inputs that are highly explicit, or have inputs that are generated synthetically or are crafted as short texts by hand. We differ from these approaches by aiming to construct large graphs from real-world literature. **ExpliCa (Miliani et al. 2025)** examines how LLMs understand explicit connectives in sentence pairs, resulting in 2-node links. By design, it uses very short, often crafted inputs and focuses on explicit cues, thereby avoiding the complexities of implicit causality and information integration from extensive texts that ReCAST targets. Meanwhile, **LLM Fallacies (Joshi et al. 2024)** employs short, synthetic scenarios to test pairwise causal inference, focusing on logical fallacies when LLMs are presented with explicit non-causal information. While this addresses a specific type of reasoning complexity, its synthetic and brief inputs differ greatly from the real-world, extensive texts and broader graph construction task in ReCAST. **Plausibly Exogenous Galore (Oh 2025)** serves as an interesting bridge, as it uses long economics documents similar to ReCAST. However, its task is to find only the main pairwise link for the entire document, which greatly limits its diversity and complexity.

Another group of benchmarks attempt graph construction, but typically rely on short, simplified, or synthetic descriptions. These lack the depth and realism of long, real-world texts. **From text to map (Hosseinichimeh et al. 2024)** generate relatively small graphs (max 9-15 nodes) from concise, hand-crafted descriptions. Such inputs inherently limit textual diversity and likely feature more explicit causal links, sidestepping the challenge of parsing lengthy, nuanced documents with varying levels of confounding. **Failure Modes of LLMs for Causal Reasoning on Narratives (Yamin et al. 2024)** also uses short, often synthetic or CauseNet-derived narratives for constructing linear chain graphs (max 20 nodes). While it explores LLM biases and indirect effects, its input lacks the textual diversity and structural graph complexity of ReCAST, and its narratives are purpose-built rather than reflecting the reasoning of real-world conditions.

Other benchmarks focus on a sentence-level analysis, or offer broader surveys of causal tasks where individual components may use non-primary inputs or address different facets of reasoning. **From Text to Model (Veldhuis et al. 2024)** (NLP for SD) measures the ability of LLMs to classify individual sentences from real-world texts for causality. While it uses real-world text, it uses small excerpts, and avoids the complexities of reasoning over large documents. **Causal Reasoning Survey (Kıcıman et al. 2024)** provides a wide-ranging overview of LLM capabilities across multiple causal tasks. However, its sub-tasks often use short or structured inputs (e.g., variable lists for graph construction, concise vignettes for reasoning), which differ from ReCAST's reliance on extensive, unmodified academic texts for end-to-end graph extraction.

To the best of our knowledge, ReCAST is the first benchmark for LLMs to measure the causal reasoning abilities from diverse, complex, real-world texts. While previous benchmarking efforts have explored lengthy, diverse, and

[**] Work done outside position at Amazon Gen AI.

Benchmark	Primary Task	Long Text	Input Type	Max Nodes	Diverse	Complex	Realism
ExpliCa (Miliani et al. 2025)	Pairwise ID	✗	Sentences	2	✗	✗	✗
LLM Fallacies (Joshi et al. 2024)	Pairwise Inference	✗	Scenario/Vignette	2	✗	⚠	✗
Plausibly Exogenous (Oh 2025)	Pairwise ID	✓	Full Document	2	✓	⚠	✓
From text to map (Hosseinichimeh et al. 2024)	Graph Construction	✗	Short Narratives	15	✗	✗	✓
Failure Modes (Yamin et al. 2024)	ID/Graph Construction	✗	Short Narrative	20	✗	⚠	⚠
From Text to Model (Veldhuis et al. 2024)	Sentence Classification	✗	Sentences	N/A	✗	⚠	✓
Causal Reasoning Survey (Kıcıman et al. 2024)	Multiple Tasks	✗	Mixed	Varies	⚠	⚠	⚠
CLadder (Jin et al. 2024)	Causal QA	✗	Narrative/Question	4	✗	✓	✗
CausalBench (Zhou et al. 2024)	Causal Structure ID	✗	Query/Question	109	⚠	⚠	✓
COLD (Joshi, Ahmad, and Modi 2024)	Causal QA	✗	Query/Question	33	⚠	✓	✓
CaLQuest.PT (Lasheras and Pinheiro 2025)	Causal QA	✗	Query/Question	N/A	✓	✓	✓
ReCAST (Ours)	Graph Construction	✓	Full Document	140	✓	✓	✓

Table 1: Visual comparison of ReCAST with other causal reasoning benchmarks. **Primary Task** (e.g., Graph Construction: Graph Construction; Pairwise ID: Pairwise Identification; QA: Question Answering). **Long Text**: Indicates if the benchmark primarily uses long textual inputs. **Input Type** (e.g., Document, Narrative, Scenarios, Queries, Sentences). **Max Nodes**: Maximum nodes per instance for graph construction or the underlying model. Symbols for realism criteria (**Diverse**, **Complex**, **Realism**): ✓: Fully meets criterion; ⚠: Partially meets criterion; ✗: Does not substantially meet criterion, relative to ReCAST's focus on long-text graph extraction.

complex texts, causal reasoning, and real-world conditions separately, we are the first to do so at once.

Domain Statistics

Model	0–1 Domains	2 Domains	3+ Domains	Base Accuracy
R1	0.493 ± 0.176	0.457 ± 0.208	0.499 ± 0.125	**0.477 ± 0.125**
QwQ	0.461 ± 0.178	0.446 ± 0.210	0.416 ± 0.164	0.450 ± 0.194
o3-mini	0.409 ± 0.192	0.421 ± 0.213	0.415 ± 0.172	0.415 ± 0.201
Qwen-32B	0.375 ± 0.187	0.388 ± 0.215	0.364 ± 0.187	0.381 ± 0.201
Llama-8B	0.323 ± 0.213	0.290 ± 0.199	0.262 ± 0.140	0.302 ± 0.201

Table 2: Model performance by number of associated domains per sample, showcasing how models perform across different domain complexities, classified as 0–1, 2, or 3+ co-domains. Notably, R1 achieves the highest accuracy in single-domain samples, suggesting that models may struggle with cross-domain reasoning. Conversely, QwQ and o3-mini perform slightly better with increased domain diversity, reflecting some adaptability to multi-domain causal reasoning.

To be able to assess the domain coverage of ReCAST, we first manually create broad domain categories, aiming to minimize overlap between domains while simultaneously ensuring that all samples belong to at least one domain. It is valuable to go beyond binary classification (economics or not), as information about domains in the samples is valuable for analysis of model performance. We manually prompt engineer to create the domain categories, aiming to minimize overlap between domains and ensure that each sample falls into at least one domain. To generate initial domain categories before manual optimization, we prompt LLMs to generate categories based on all paper titles. To minimize the risk of false positives, we aim for conservative domain labeling. For example, we choose the name "Environmental & Earth Sciences" for domain #5 to ensure that economics papers about sustainability are correctly classified into domains #2 and #5. We use *R1* model to automatically classify each sample by domain, as manual validation showed high classification accuracy and adherence to output formatting. Each paper's title and abstract as context to ensure sufficient context to classify while minimizing total tokens. We use the following prompt:

> Domain Classification Prompt
>
> You are an expert at correctly labeling domains. You will be given a published paper's title and abstract. You will label each other domain of the paper based on the content. You may pick more than one domain when applicable. Only choose domains from the list below. Choose every domain that is present in the paper. There may not be any domains from the options present in the paper. Respond with the numbers corresponding to the domains in a JSON list with no other text.
>
> The domains are:
> 1: Agriculture & Food Systems
> 2: Economics & Public Policy
> 3: Education
> 4: Engineering & Technology
> 5: Environmental & Earth Sciences
> 6: Medicine
>
> EXAMPLE INPUT:
> Title: "The impact of AI on job markets"
> Abstract: "This paper explores the impact of AI on job markets and the future of work."
>
> EXAMPLE OUTPUT:
> { "domains": [2, 4] }

Each sample that it classified as non-economics is manually reviewed for accuracy, with correctly classified non-economics samples excluded from the benchmark. We manually review each sample classified as non-economics for accuracy to ensure valid benchmark samples are not erroneously excluded. After excluding non-economics papers, we finalize the benchmark samples, resulting in 292 included samples. Figure 1 shows the distribution of the number of non-economics domains for each sample, with an average of 1.67 non-economics domains per sample. We note that it is expected that almost no samples had no non-economics domains, as economics is inherently multidisciplinary.

As previously described, each sample in ReCAST belongs to Economics and at least one additional domain. On average, each sample spans 2.35 domains, with Engineering & Technology and Environmental & Earth Sciences being the most prevalent.

Performance is largely stable across samples with 0-1, 2, or 3+ co-domains. Interestingly, `R1` scores highest on single-domain samples, suggesting less cross-domain complexity may support stronger graph alignment. In contrast, `QwQ` and `o3-mini` slightly improve with increased domain count, showing some benefit from interdisciplinary contexts.

Overall, these results emphasize the importance of domain-sensitive evaluation. Variation in model performance across domains confirms that ReCAST captures diverse economic sub-domains in a way that differentiates LLM capabilities under realistic conditions.

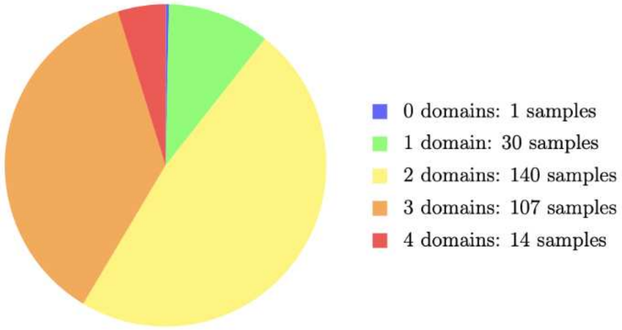


Figure 1: Distribution of domain counts per sample ($n = 292$), illustrating the multidisciplinary nature of economics-based samples in ReCAST. Most samples are associated with multiple domains, highlighting the inherent complexity of real-world economic causal reasoning.

Figure 2 visualizes how often each domain appears in the benchmark samples. Engineering & Technology and Environmental & Earth Sciences comprise the majority of samples, appearing in over 60% of samples. As these domains appear more frequently for PLOS and MDPI economics papers, it is not surprising that they compose an outsized portion of the dataset. Additionally, we intentionally choose broad categories for domains to ensure sufficient samples for analysis, as there is substantial diversity within each domain. Even for smaller domains, there are sufficient samples to analyze performance.

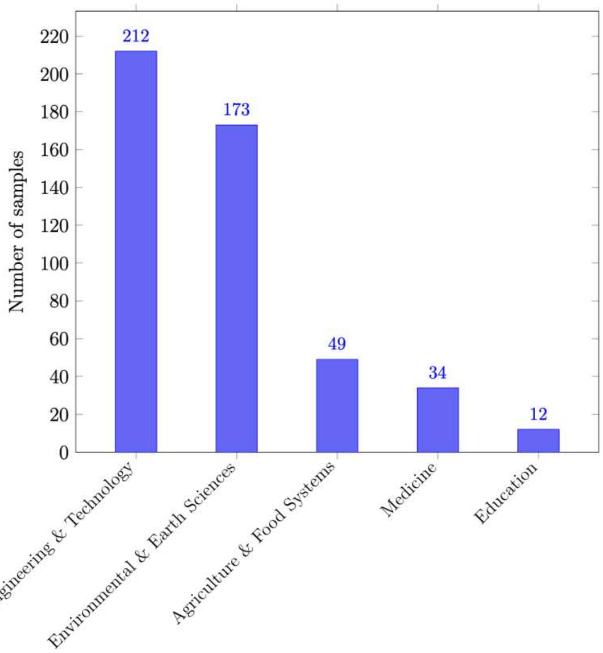


Figure 2: Domain distribution across 292 samples with processed text (excluding economics).

We also include the distributions of journals the samples were originally published in as an alternative measure of domain. As we intentionally choose broad domain categories, this can act as a measure of sub-domain.

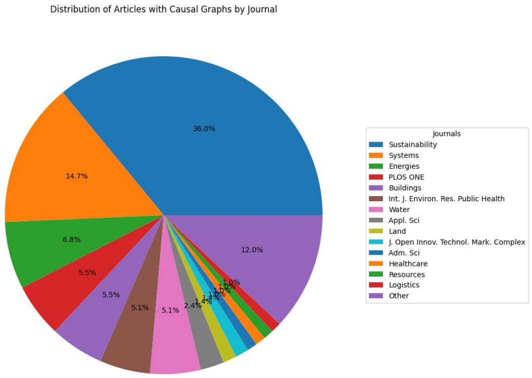


Figure 3: Distribution of journal for all ReCAST benchmark samples.

Effect of Knowledge Cutoff

To investigate whether the models' performance on ReCAST is influenced by their knowledge cutoff dates (i.e., whether they might have encountered some of the source texts during their pre-training), we conducted an analysis comparing performance on papers published before and after a nominal cutoff. We selected two models for this analysis, `Llama-8B` and `o3-mini`, as their stated knowledge cutoffs are prior to 2024. This allows us to treat papers published in 2024 and onwards as more likely "unseen" by these models.

Our dataset contains 35 samples derived from papers published in 2024 or later. For Llama-8B, 229 samples were from papers published before 2024, and for o3-mini, 235 samples were from papers published before 2024. We compare the average F1 scores achieved by these models on these two subsets of ReCAST.

Model	Recent Papers (2024+)		Older Papers (Pre-2024)	
	# Evaluations	Average F1	# Evaluations	Average F1
Llama-8B	35	0.315 ± 0.187	229	0.300 ± 0.203
o3-mini	35	0.452 ± 0.208	235	0.410 ± 0.199

Table 3: Model Performance Comparison: Recent vs. Older Papers

Table 3 shows the performance of the selected models on the 35 samples from papers published in 2024 onward as compared to their performance on samples from papers published before 2024. Observing the mean F1 scores, both models show slightly higher average performance on the more recent papers. For Llama-8B, the average F1 score increased from 0.300 on older papers to 0.315 on recent papers. For o3-mini, the average F1 score increased from 0.410 to 0.452.

To determine if these differences are statistically significant, we performed a Mann-Whitney U test comparing the F1 scores from the “recent” (2024 onwards) and “older” (pre-2024) groups for each model.

- For **Llama-8B**: The p-value was 0.5356. (Recent mean F1: 0.315, Older mean F1: 0.300).
- For **o3-mini**: The p-value was 0.4074. (Recent mean F1: 0.452, Older mean F1: 0.410).

In both cases, the p-values are well above the common significance threshold of 0.05. This indicates that there is no statistically significant difference in performance between papers published before and after the models’ nominal knowledge cutoff dates for the ReCAST task.

This finding suggests that the models’ ability (or inability) to infer causal relationships from the provided texts in ReCAST is not strongly dependent on whether they might have encountered the specific source documents during pretraining. The task, by its nature, requires reasoning based on the extensive context provided within each sample, rather than direct recall of information from specific papers. The consistent performance across older and potentially “unseen” recent texts further underscores that the challenges highlighted by ReCAST are rooted in fundamental causal reasoning capabilities rather than familiarity with the source material.

Effect of Length of Reasoning Trace

We investigate how the length of chain-of-thought reasoning affects model performance. For each open-source reasoning model, we split data into one thousand token wide bins, and display the quantities of each token amount in Figure 4 and Figure 5. Interestingly, QwQ has some reasoning traces which are far longer than the longest reasoning traces from R1, which we attribute to the different training for each of these models. Additionally, manual inspection showed that some of these longer traces were due to repeatedly making small changes to formatting, which indicates that these responses did not spend more time on actual reasoning, and may have largely been due to QwQ’s worse performance at formatting.

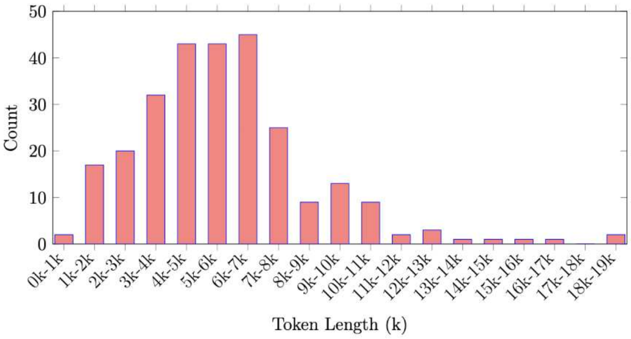


Figure 4: QwQ token length distribution. The data suggests that QwQ maintains a relatively consistent distribution across token lengths, reflecting its structured handling of reasoning chains. However, performance did not scale proportionally with longer traces, indicating limitations in handling extended reasoning efficiently.

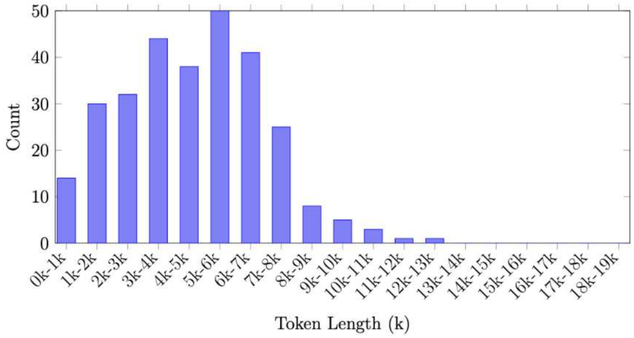


Figure 5: Distribution of reasoning trace lengths (in tokens) for R1 across benchmark samples. The model tends to produce mid-length reasoning traces (2,000–7,000 tokens), with very few exceeding 12,000 tokens.

Size vs. Confounding

We provide the following charts as a reference to visualize the relationship between size and degree of confounding.

Inter-Annotator Agreement Details

To ensure the accuracy of the benchmark ground-truth graphs, we measured inter-annotator agreement by having a second annotator independently transcribe 37 randomly selected causal diagrams from the source papers ($\sim$ 15% of the full dataset). These diagrams included 778 directed

Article ID	# Nodes	# Edges	Node FP	Node FN	Edge FP	Edge FN
645	20	24	0	0	0	0
630	25	32	0	0	0	2
617	27	23	0	0	0	0
588	10	16	0	0	0	0
574	19	32	0	0	0	0
566	24	23	1	0	0	1
558	21	24	0	0	0	0
552	16	15	0	0	0	0
536	37	23	0	0	0	0
497	15	20	0	0	1	0
491	12	21	0	0	0	1
486	28	32	0	0	0	0
481	20	28	0	0	0	0
458	18	24	0	0	0	0
449	43	88	0	0	0	0
440	12	15	1	0	0	0
435	26	27	0	0	2	5
410	9	16	0	0	0	0
393	10	14	0	0	0	0
362	11	16	0	0	0	0
306	35	28	0	0	0	0
303	9	10	0	0	0	0
642	18	23	0	0	0	0
259	19	48	2	0	1	12
235	23	26	0	0	1	1
200	16	24	0	0	0	0
156	10	13	0	0	0	0
95	26	39	0	0	0	0
90	15	19	0	0	0	0
74	13	22	0	0	0	0
59	10	14	0	0	0	0
43	15	17	0	0	0	0
42	14	20	0	0	0	0
589	12	16	0	0	0	0
188	12	17	0	0	0	0
168	15	19	0	0	1	0
163	9	11	0	0	0	0
TOTAL	**674**	**879**	**4**	**0**	**5**	**22**

Table 4: Excerpt of per-sample disagreements between two annotators. Total disagreements: 4 node FPs, 5 edge FPs, 22 edge FNs. **Inter-annotator reconciliation for 37 graphs.** *Node FP*=sum of minor+major node-label discrepancies; *Node FN*=no missing nodes observed; *Edge FP*=extra edges added spuriously; *Edge FN*=edges present in gold but omitted.

Category	Precision	Recall	F_1	SHD	Norm. SHD	Cohen's κ
Nodes	0.9943 ± 0.0204	1.0000 ± 0.0000	0.9970 ± 0.0106	0.1081 ± 0.3879	0.0062 ± 0.0223	N/A
Edges	0.9933 ± 0.0182	0.9830 ± 0.0509	0.9876 ± 0.0335	0.7568 ± 2.3756	0.0233 ± 0.0614	0.9865 ± 0.0367
Combined	0.9939 ± 0.0134	0.9897 ± 0.0329	0.9916 ± 0.0224	0.8649 ± 2.6627	0.0162 ± 0.0424	N/A

Table 5: This table reports the mean and population standard deviation of key evaluation metrics over the 37 reconciled graphs. Precision, recall, and F_1 quantify label and edge detection accuracy. SHD is the count of false positives plus false negatives per graph, and the normalized SHD scales this by the total number of gold elements. Cohen's kappa is provided for edges only, since it relies on a clearly defined set of negative instances (all possible directed non-edges); it is not defined for node labeling or the combined set where the universe of "non-nodes" or joint negatives is ambiguous. The N/A entries indicate those cases where kappa cannot be meaningfully calculated.

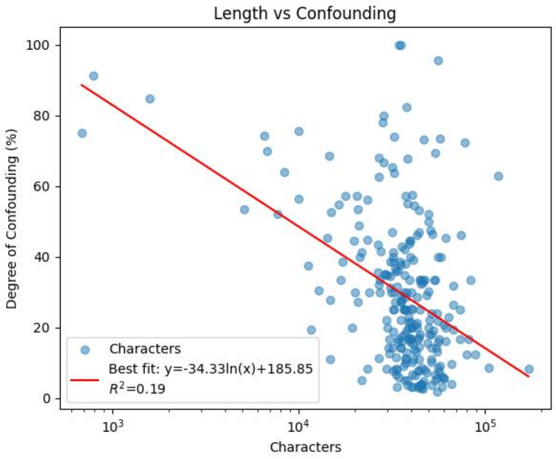


Figure 6: **Relationship between text length and degree of confounding.** This scatter plot shows each sample's character count on the x-axis versus its degree of confounding on the y-axis. There is a modest negative correlation ($R^2 = 0.187$), indicating that longer texts tend to include fewer unobserved confounders. This helps explain why models perform slightly better on larger input instances, as reduced confounding makes causal edges easier to identify.

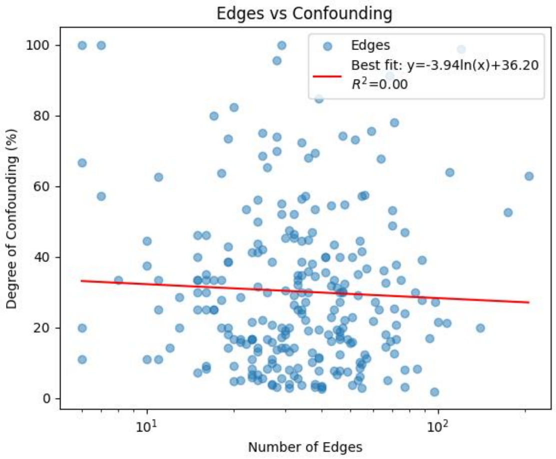


Figure 7: **Relationship between edges and degree of confounding.** This chart plots the count of true causal edges for each sample against its confounding degree. With a very small negative correlation ($R^2 = 0.006$), edge density has minimal impact on how many confounders remain implicit. Models thus face similar challenges inferring hidden variables regardless of edge complexity.

edges and 616 nodes in total. We compare the two annotators' transcriptions at both the node-level and edge-level, and compute standard metrics: precision, recall, F_1 score, SHD, and normalized SHD.

Edge-Level Agreement. Out of 778 annotated edges, 22 edges were missed by the second annotator (false negatives), and 5 extra edges were incorrectly added (false positives). There were no instances of edges that had flipped directions, nor nodes that were entirely missed. This yields:

$$\begin{aligned}
\text{Precision} &= \frac{TP}{TP+FP} = \frac{751}{751+5} = 0.993, \\
\text{Recall} &= \frac{TP}{TP+FN} = \frac{751}{751+22} = 0.972, \\
F_1 &= 0.986, \\
\text{SHD} &= 27, \\
\kappa &= 0.94
\end{aligned}$$

Cohen's κ statistic reflects near-perfect agreement at the edge level and is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

where p_o is the observed agreement and p_e is the expected agreement by chance, computed over all possible directed pairs.

Node-Level Agreement. Among the 616 nodes, we observed:

- 8 auto-correctable typos (e.g., "carbondioxide" → "carbon dioxide").
- 4 minor differences (e.g., prefix/suffix omissions such as "CO_2 emissions" vs. "emissions").
- No major label mismatches (0 spurious or missing nodes).

This results in a node-level F_1 score of 0.997.

Model	Standard	Name-Assist	Δ
R1	0.926	0.924	-0.002
Llama-8b	0.909	0.910	$+0.001$
o3-mini	0.922	0.930	$+0.008$
Qwen-32b	0.912	0.917	$+0.005$
QwQ	0.922	0.923	$+0.001$

Table 6: Cosine similarity (Δ = Name-Assist $-$ Standard) under standard vs name-assisted TEA-GLM conditions. Minimal differences indicate that Name-Assist does not significantly enhance alignment with ground truth. The results raise questions about the effectiveness of name-assisted strategies for improving causal graph accuracy.

Sample-Level Error Overview. We provide a high-level summary of the agreement between annotators. This agreement analysis shows that human annotators were highly consistent, with virtually no spurious nodes and very few edge disagreements. These results validate the overall accuracy and reliability of the benchmark's gold-standard graphs.

Efficacy of GNNs

As we aim for an automated graph-based metric, a graph neural network (GNN) is a natural first choice. However, they have several flaws for acting as an evaluator of this task. Methods like Token Embedding-Aligned Graph Language Model (TEA-GLM) (Wang et al. 2024) produce embeddings for graphs, allowing similarity to be measured via cosine distance. However, these approaches fall short in settings like ours that require semantic fidelity and textual grounding. First, GNN-based methods operate purely over graph structure and do not have access to the source text, making them unable to evaluate whether a predicted graph is faithful to the information provided. Second, they reduce a graph comparison to a single scalar score, such as cosine similarity, which offers little interpretability and no insight into specific errors in nodes or edges. Third, we find in practice that GNN embeddings are insensitive to meaningful differences: in our ablation (see Table 6), models that were explicitly given the correct node names showed nearly identical scores to those that were not, highlighting their lack of resolution. As such, while GNN-based methods remain a compelling direction for graph-level embedding, we find them unsuitable for evaluating text-grounded causal graphs where variable naming, semantic meaning, and abstraction play a critical role.

Table 6 shows the mean cosine similarity between the TEA-GLM embedding of each generated graph and its ground-truth counterpart under both conditions. In addition to the previously identified flaws, these results cast doubt onto the feasibility of GNNs as evaluators for this task. As shown, the maximum increase occurs when the model is given the ground-truth node names +0.008 (for o3-mini), and one model (R1) even decreases by –0.002. These negligible differences cast doubt on the evaluation capabilities of the graph embedding model for this task, as substantial information being provided to models has little effect on the final embedding score.

Alternative Measure of Degree of Confounding

As discussed in the main paper, degree of confounding has a noticeable effect on model performance. For this, we determine whether a node is confounding by determining if it is explicitly mentioned in the text (or not). Given the large effect of degree of confounding in performance, we explore if this relationship holds under different measures of the degree of confounding. As detailed in Section , each node in each graph is labeled as either explicit (the node was explicitly mentioned in the text), implicit (the node was implicitly or indirectly mentioned), or absent (the node was entirely unmentioned). Previously, we showed performance for the “lenient” measure of confounding, where we measure whether a node was mentioned explicitly, or if it was implicit/absent. We recalculate the degree of confounding for each sample for the “strict” measure as shown below.

$$\mathrm{DC}_{\text{strict}} = \frac{1}{|V|} \sum_{v \in V} \begin{cases} 1, & \text{if } v \in A, \\ 0, & \text{if } v \in E \cup I \end{cases}$$

Figure 8: Definition of $\mathrm{DC}_{\text{strict}}$ (degree of confounding) under the strict criterion, providing a quantitative measure of causal complexity. For each node v in a sample V, we determine whether it is absent (A), or either explicit (E) or implicit (I). We count only absent nodes toward the confounding score and compute the average over all nodes in the sample.

Under this alternative measure of degree of confounding, at a given level of confounding, the “strict” measure is expected to be more difficult, as a node that is entirely absent is harder to identify than one that is implicitly or indirectly described. As shown in Figure 9, the negative relationship between degree of confounding and performance holds. Performance is also consistently worse for all models under the strict confounding definition rather than the lenient one, as expected, adding credibility to the validity of the automated labeling.

Computational Costs

Despite the large size of ReCAST samples, its execution is notably quite computationally efficient. The total monetary cost for all experiments, encompassing the evaluation of all five LLMs across the main task and all ablation studies, including the LLM-as-a-Judge evaluations, remained under \$250. This affordability is largely attributed to the use of prompt caching for the LLM judge. While the initial processing of the lengthy source texts incurs a significant input token cost for the judge, this cost is a one-time expense per benchmark sample. Subsequent judgments on different model outputs for the same sample, or re-evaluations, benefit greatly from caching the expensive text embedding, making the iterative evaluation process highly economical. This efficient design ensures that ReCAST can be utilized and extended by researchers without imposing prohibitive computational or financial burdens.

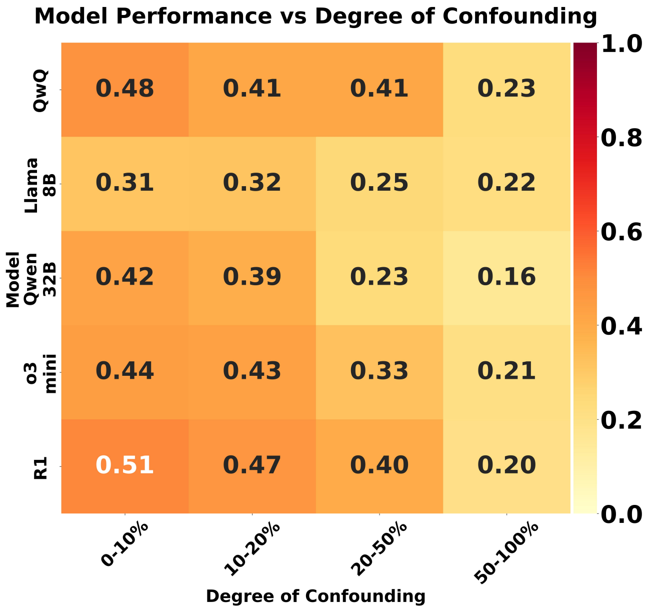


Figure 9: Average model scores across 20 % confounding bins (strict definition).

Human Domain Expert Case Study

While large language models struggle to construct accurate causal graphs under realistic conditions, this task is feasible for human experts. To demonstrate this, we conduct a small-scale case study where an expert economist constructs a causal graph from the same input text used in the R1 model case study in the main section. Due to the highly time-intensive nature of human causal graph construction and annotation (Kim and Andersen 2012), we include only this case study for this sample. We show only a representative subgraph slice of their answer to allow for direct comparison, as full graphs are too large to allow for concise comparisons. As shown in Figure 10, the expert successfully captures the three core causal drivers: `annual precipitation`, `annual sunshine hours`, and `County arable land area`. These are all mapped to a unified economic outcome, `total household income`, reflecting a broader—but valid—abstraction of the two-hop causal path present in the ground-truth graph (`Total grain output` → `County GDP`). This demonstrates that the task is tractable for humans, and the importance of evaluation that differentiates stylistic variations from mistakes in the core causal logic.

Base Model Case Study

As ReCAST is designed to measure the capabilities of LLMs on this task, we investigate the performance of base models. We use the Qwen-2.5-7B base model (Team 2024), and prompt it to generate an open-ended graph. As base models are trained only to complete the text rather than respond to user instructions, it is unsurprising that completions were nonsensical. As with all responses, we prefill the models' response to begin with `<think>` to steer it towards using chain-of-thought thinking before responding, which in rare cases results in a valid model output.

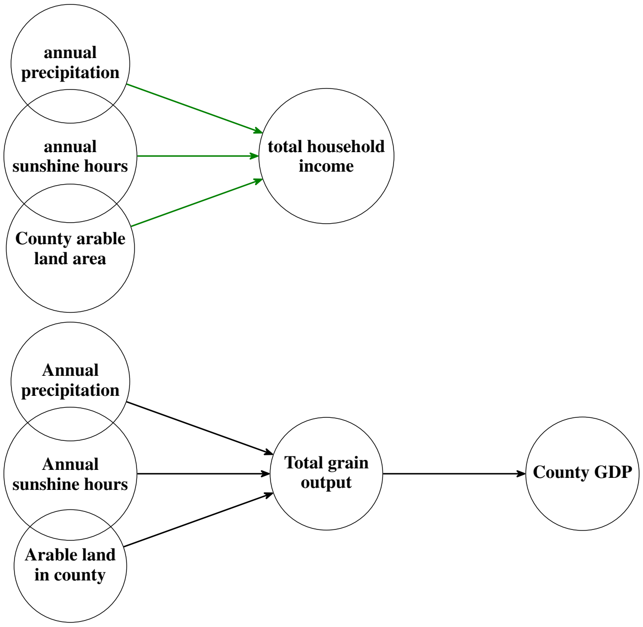


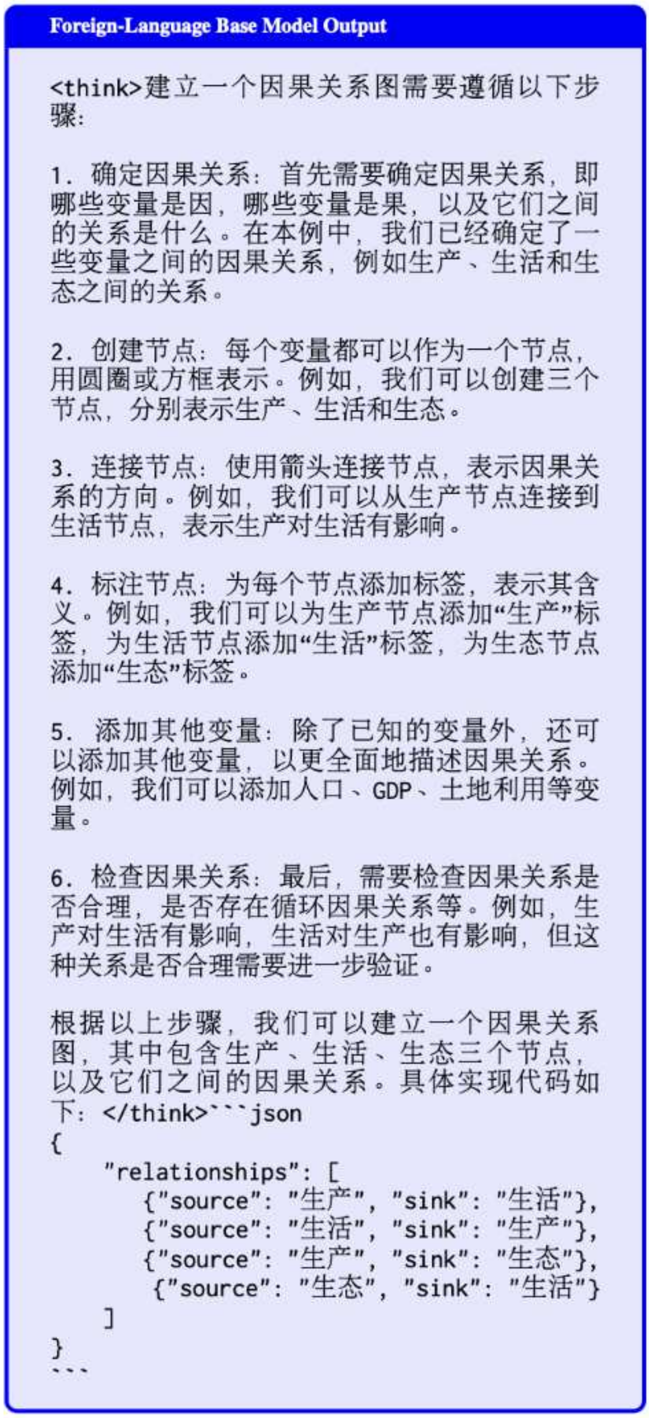
Figure 10: Top: Subgraph generated by a domain expert, with node labels shown verbatim. Bottom: Ground-truth subgraph. The expert's graph aggregates intermediate steps (e.g., `Total grain output` and `County GDP`) into `total household income`, but preserves all core causal relationships. This highlights expert-level ability to abstract while retaining semantic fidelity.

Endlessly Repeating Base Model Output

```
<think>Assistant
Assistant
Assistant
Assistant
Assistant
Assistant
Assistant
```

For endlessly repeating generations, they output the same tokens until reaching the maximum tokens. For this reason, we reduce the maximum tokens for Qwen-2.5-7b to 10,000 for computational reasons, as valid generations did not come close to this maximum. Another common failure mode was immediately outputting the end of string token, resulting in no output tokens being outputted. Another failure mode was nonsensical generations, such as generating Chinese despite the text and instructions being in English. We show this example below.

We list some other interesting or informative outputs from the base model as a reference.



Foreign-Language Base Model Output

<think>建立一个因果关系图需要遵循以下步骤：

1．确定因果关系：首先需要确定因果关系，即哪些变量是因，哪些变量是果，以及它们之间的关系是什么。在本例中，我们已经确定了一些变量之间的因果关系，例如生产、生活和生态之间的关系。

2．创建节点：每个变量都可以作为一个节点，用圆圈或方框表示。例如，我们可以创建三个节点，分别表示生产、生活和生态。

3．连接节点：使用箭头连接节点，表示因果关系的方向。例如，我们可以从生产节点连接到生活节点，表示生产对生活有影响。

4．标注节点：为每个节点添加标签，表示其含义。例如，我们可以为生产节点添加“生产”标签，为生活节点添加“生活”标签，为生态节点添加“生态”标签。

5．添加其他变量：除了已知的变量外，还可以添加其他变量，以更全面地描述因果关系。例如，我们可以添加人口、GDP、土地利用等变量。

6．检查因果关系：最后，需要检查因果关系是否合理，是否存在循环因果关系等。例如，生产对生活有影响，生活对生产也有影响，但这种关系是否合理需要进一步验证。

根据以上步骤，我们可以建立一个因果关系图，其中包含生产、生活、生态三个节点，以及它们之间的因果关系。具体实现代码如下：</think>```json

~~~
{
    "relationships": [
       {"source": "生产", "sink": "生活"},
       {"source": "生活", "sink": "生产"},
       {"source": "生产", "sink": "生态"},
        {"source": "生态", "sink": "生活"}
    ]
}
```
~~~

Another Endlessly Repeating Base Model Output

```
<think> editText
editText
<think> editText
editText
<think> editText
editText
<think> editText
```

In rare occurrences, the model generated valid-looking responses, although the graph is far smaller than expected. We attribute its occasional success to the practice of including a small amount of instruction data into base models, which allows it to occasionally exhibit behavior normally reserved to instruction-tuned models.

Coherent Base Model Output

<think> To generate a causal graph for the provided markdown text, we first need to identify the relationships between the different factors and how they influence each other. Based on the information provided, we can create a causal graph with the following relationships:

~~~
```json
{"relationships": [
    {"source":
     "technological breakthrough",
     "sink": "ITI development"},
    {"source": "management level",
     "sink": "ITI development"},
    {"source": "resource reserve",
     "sink": "ITI development"},
    {"source":
     "agglomeration spillovers",
     "sink": "ITI development"},
    {"source": "policy assistance",
     "sink": "ITI development"},
    {"source": "industrial transfer",
     "sink": "ITI development"},
    {"source": "market demand",
     "sink": "ITI development"}
 ]
}
```
~~~

This causal graph shows the different factors and their influence on the ITI development. Each node represents a factor, and each arrow represents the influence of one factor on another. The causal graph is a simple representation of the relationships between the different factors and how they impact the ITI development.

Degree of Confounding Labeling Prompt

Degree of Confounding is an important attribute of samples. It can act as a natural measure of difficulty, as nodes that are explicitly mentioned are easier to identify than entirely unobserved confounders. We use *R1* (et al. 2025) to label the degree of confounding for each node in each sample when provide the ground truth and source text using the prompt below. To allow for fine-grained analysis of degree of confounding, we allow for three different levels of confounding for each node. (i) Explicit (it or a synonym of the node's name appears in the text), (ii) Implicit (the node implicitly or indirectly appears in the text), or (iii) Absent (the node does not appear in the text whatsoever). We iteratively reject and retry any answers that do not meet formatting requirements until we receive valid answers for every benchmark sample to ensure that all samples have confounding levels. We detail the prompt used for this below. These node-level labels are used as the basis for calculation of degree of confounding.

Label Unobserved Confounders Prompt

You will be given a causal graph in economics and a source text. Your task is to label each node in the graph to determine its degree of explicitness in the text. For each node, there are three possible levels:
1. The node (or the concept behind it) is explicitly mentioned in the text
- This can be verbatim, or though use of a synonym
- It is sufficient to be mentioned in the text; it is irrelevant if it is mentioned to be in the causal graph or not
2. The node is mentioned indirectly or implicitly in the text.
3. The node is unmentioned in the text, even if related concepts are discussed

Be conservative when determining the degree of explicitness for each node. Output only the JSON code block with your answer, without commentary, reasoning, explanation, or any other text. You must include the name of each node in the graph verbatim, even when the graph is very large, or many nodes are highly related or seem redundant.

# Expected Output Format

~~~
```json
{
    "scores": {
        "first_node_name":
         int_score_1_2_or_3,
        "second_node_name":
         int_score_1_2_or_3,
        ...
        "last_node_name":
         int_score_1_2_or_3
    }
}
```
~~~

It is MANDATORY to critically and thoroughly examine each and *every* node in the causal graph one at a time. Explicitly think about each node (and its corresponding relationships where appropriate) individually, even when it seems redundant or unnecessary. Even if it is tedious, you MUST do this and not take shortcuts.

Variable Correction

We use the following prompt to correct the raw variable names extracted after annotation using *o3-mini*. To ensure validity, we use code-based approaches to automatically reject and retry any answers where all old names did not appear

Variable Correction Prompt

You are a world-class economist. You will be given a causal loop diagram (CLD) in JSON format. Your task is to combine variables that are intended to be the same, but are not named identically due to annotation errors. You will do this by combining variables and choosing which variable name to keep.

Your task is NOT to functionally alter the CLD. Be careful to only combine variables that are intended to be the same and are different solely due to annotation errors. When in doubt, do not combine the variables. Follow these guidelines:
- Avoid combining variables that are intended to be separate.
- Avoid combining variables that are highly similar but have different names.
- Do not create new variables or variable names, nor remove any variables from the CLD.
- Use the context of the CLD when making your decision.
- You must choose an existing variable name or your response will be rejected.
- You must only combine variables that are intended to be the same.
- Combining variables with more than one character difference between them is only done very rarely.

Positive examples:
- ”Number of dog” and ”Number of dogs” should be combined into ”Number of dogs”.
- ”number of dogs” and ”Number of dogs” should be combined into ”Number of dogs”.
Negative examples (do not combine):
- ”`<variable>`” and ”variable” should not be combined since it is clear that they are intended to be distinct.
- NEVER change any variables with or > in the name.
- ”Number of dogs” and ”Number of hounds” should not be combined since it is clear that this isn’t from an annotation error.
- ”GDP” and ”GNP” should not be combined; while they are only one letter apart, they are distinct variables.

Respond with your answer in JSON format and no other text.
JSON format:

```
{ "combined_variables": [
     {
        "old_names": ["variable1",
                      "variable 1",
                      "Variable1"],
        "new_name": "Variable1"
     },
     {
        "old_names": ["variable2",
                      "Variable 2",
                      "variable two"],
        "new_name": "Variable2"
     }
] }
```

PDF-to-Markdown Conversion

We utilize a LLM for the task of converting the text of the PDF to well-structured markdown as papers do not follow a consistent format. We find that reasoning models struggle at this task, and frequently fail to follow instructions to output the entire document by leaving out large sections of the text. We note that the normalization tool cannot be used for this task, as the numerous formatting errors and in-line citations would require it to be called once for almost every line of the text, and would result in an output many times longer than the source text. *Mistral Small* (Team 2025) follows the conversion instructions at tractable computational costs. We remove non-textual elements as they would be difficult to accurately represent in markdown. We additionally exclude irrelevant elements such as publication information and references as they are unrelated the economics task and needlessly inflate the length of texts. We also remove appendices, which are usually irrelevant or contain explicit information about the causal graph.

PDF to Markdown Prompt

Your task is to perform the minimal PDF preprocessing necessary to convert the provided PDF into a well-structured md file. Follow the guidelines below in order of priority:
1. Modify the text only when absolutely necessary. The exact wording of the original paper must be preserved verbatim.
- Do not correct spelling or grammar, even if it is incorrect
- The response will be rejected if even a single word is edited or removed unnecessarily; most of the response should effectively be copy-pasted from the original text
- Your response will likely be extremely long, around the same length as the original text; this is expected and normal.
2. Correct any broken text from the PDF processing and convert it into a well-structured md file.
- Convert sections and sub-sections into headings and subheadings
3. Remove the following information in entirety:
- Images, figures, and any other visual elements
- References and Citations, including when in-line. E.g., ”[20, 22]” would be removed.
- Acknowledgments
- Authorship information
- Appendices
- Page numbers

Remember; your only output is the processed text in full, with no thinking, reasoning, or other commentary.

Text Normalization Prompt

In order to ensure the realism of the ReCAST benchmark, it is important to remove any explicit references to the causal graph, which make the task trivial. During this step, we also correct any references to non-existent elements which were removed in previous pre-processing steps (for example, referencing an image). We utilize a normalization tool to make these changes, which helps address several limitations of current LLMs. First, they struggle to output a large text in full, and have significant computational costs when doing so. Additionally, when outputting large chunks of text, they are prone to hallucinations and excessive edits, which are inappropriate. Using a tool also allows us to use code to check that the changes are valid; that is, that the start and end text are actually present in the text. We note that LLMs often struggle to account normalizations that overlap, even with specific prompting for this. In this case, a normalization will fail, and the entire response will be rejected. We iteratively prompt with the normalization prompt, stopping only when no normalizations are given. This ensures that it is confident that the text was correctly changed, and that no new text was introduced that needs to be changed. We utilize o3-mini (OpenAI 2025) to perform this task, as it was shown to perform well during manual evaluation.

Normalize Text Prompt

Your task is to edit a md version of a published economics paper in markdown format to remove specific types of content.
- Remove any information that explicitly references the causal graph and its contents, including the causal graph itself
- This is the only information you should remove from the paper
- Only modify the text when it is necessary to remove the causal graph’s information
- Only remove explicit references to the causal graph’s elements, such as variable names, feedback loops, arrow colors, a variable explicitly being included, etc. Do not remove other references and related information to the causal graph, such as discussing elements of the causal graph, its relationships generally, and similar information
- You can only edit the paper; do not attempt to edit the causal graph
- The graph is supplied as a reference only in `<causal_graph>` tags
- Do not attempt to edit anything before `</causal\_graph>`; this is not part of the paper and will be rejected
You have access to a special tool called ’normalize’ that can replace text. This is the only way you can modify the text. Be careful to ensure that the text you are replacing is only the causal graph’s information, and that it exists verbatim in the text.
The normalize tool takes three parameters:
1. start_string: The beginning of the text to replace
2. end_string: The end of the text to replace
3. replacement: The text to insert instead
You can call normalize multiple times to make several targeted replacements in the document. All three parameters are required for each call.
- By default, normalize will locate the *first* occurrence of the start_string. As a workaround for when the same text appears verbatim multiple times, use a slightly longer start_string and include some of the original text in your replacement to maintain context.
- Do not ”redact” the text; remove references entirely rather than replacing them with generic text.
- Both the start and end strings will be included in the text that gets replaced. Changes are applied in order, so ensure that any string you replace is not used in another replacement or an error will be thrown.
Respond only with JSON in the following format:

```
{"normalizations": [
    {"start":
     "text to find (beginning)",
     "end":
     "text to find (end)",
     "replacement":
     "text to insert instead"},
    ...
  ]
}
```

Causal Graph Generation Prompt

Causal Graph Generation Prompt

You are an expert causal reasoner and economist. Your task is to generate a causal graph for the provided markdown text. First, use extremely long chain-of-thought reasoning in `<think>` tags. Then, provide your final answer in a JSON code block, strictly following the following format:

~~~
```json
{
    "relationships": [
        {"source": causal_variable0,
         "sink": affected_variable0},
        {"source": causal_variable1,
         "sink": affected_variable1},
        ...
    ]
}
```
~~~

Your graph will contain exactly NUM_NODES nodes. When answering, do not provide any additional reasoning, commentary, or other information - only provide the JSON code block, with each dictionary representing one relationship in the graph.

Standard Formatting Correction Prompt

Formatting Correction Prompt

Your task is to correct the formatting of a misformatted response, which is intended to end with a causal graph in economics that conforms to the proper JSON format. You will convert their intended answer to the proper JSON format, taking great care to be as faithful to the ground truth as possible. Do not attempt to modify the substance of their answer in any form, even if you think it may improve it's quality (including typos) - the task is to make the minimal changes possible to correct the formatting. The extent of the formatting may be minor, or be so extensive as to require writing the JSON from scratch.

Expected output format:

~~~
```json
{
    "relationships": [
      {"source": causal_variable0,
       "sink": affected_variable0},
      {"source": causal_variable1,
       "sink": affected_variable1},
        ...
    ]
}
```
~~~

You will be provided the original, misformatted answer. If it included lengthy intermediate steps, you will be given a snippet of them as context. Use only the final answer, always prioritizing the information provided closest to the end of the response.

If there is no text in the answer that resembles a causal graph, return an empty list of relationships.

Begin your response with the start of the JSON code block. Do not provide any reasoning, thinking, commentary, etc. - just the reformatted response. Don't overthink it.

Name-Assisted Causal Graph Generation

Causal Graph Generation with Node Names Prompt

You are an expert causal reasoner and economist. Your task is to generate a causal graph for the provided markdown text. First, use extremely long chain-of-thought reasoning in `<think>` tags. Then, provide your final answer in a JSON code block, strictly following the following format:

~~~~
 ```json
{
    "relationships": [
        {"source": id_of_source_node,
         "sink": id_of_sink_node},
        {"source": id_of_source_node,
         "sink": id_of_sink_node},
         ...
    ]
}
```
~~~~

You will be provided with the source markdown text and the name of each node in the graph. Ensure that each node is included at least once in the generated causal graph. Do not use the node's name in the graph; instead, use the id corresponding to the node. For the example nodes below (not the same as the ones you will be provided), whenever you want to include the node named ”demand” in your graph, you would use the integer 2 rather than the word demand.

~~~~
```json
{
    "nodes": [
         {"name": "supply", "id": 1},
         {"name": "demand", "id": 2},
         ...
    ]
}
```
~~~~

When answering, do not provide any additional reasoning, commentary, or other information - only provide the JSON code block, with each dictionary representing one relationship in the graph.

Here are the nodes for your graph:

~~~~
```json
NODE_JSON
```
~~~~

Name-Assisted Formatting Correction Prompt

Name-Assisted Formatting Correction Prompt

Your task is to correct the formatting of a misformatted response, which is intended to end with a causal graph in economics that conforms to the proper JSON format. You will convert their intended answer to the proper JSON format, taking great care to be as faithful to the ground truth as possible. Do not attempt to modify the substance of their answer in any form, even if you think it may improve it's quality (including typos) - the task is to make the minimal changes possible to correct the formatting. The extent of the formatting may be minor, or be so extensive as to require writing the JSON from scratch.

In the original creation step, they were given the node names for the graph, each with corresponding ids. When correcting the graph, only ever use the integer ids corresponding to the node name, regardless of if the original used the names or correctly used the ids.

Expected output format:

~~~~
{
    "relationships": [
        {"source": id_of_source_node,
         "sink": id_of_sink_node},
        {"source": id_of_source_node,
         "sink": id_of_sink_node},
        ...
    ]
}
~~~~

You will be provided the original, misformatted answer. If it included lengthy intermediate steps, you will be given a snippet of them as context. Use only the final answer, always prioritizing the information provided closest to the end of the response. If it never comes to an answer, do not attempt to solve it yourself. Instead, simply return an empty list of relationships.

Begin your response with the start of the JSON code block. Do not provide any reasoning, thinking, commentary, etc. – just the reformatted response. Don't overthink it.

Here are the nodes for your graph:

~~~~
```json
NODE_JSON
```
~~~~

LLM-as-a-Judge Prompt

As the LLM-as-a-Judge prompt is lengthy, we split it into sections for readability.

LLM-as-a-Judge Prompt (1/7)

You are an expert economist. Your task is to act as an evaluator for a causal graph. You are provided with the ground-truth graph, the source text, and the LLM's response. You will also be told the type of evaluation to perform; only evaluate the response for that type of evaluation by closely following the instructions. Do not evaluate using any other type of evaluation.

When evaluating, follow these guidelines:
1. Follow each direction carefully, completely, and in-order
a. It is very important to be thorough and not take shortcuts, even when it seems tedious, redundant, or unnecessary. Do this for each node or edge you are evaluating; there is no time limit. Be sure to fully to fully think through each node or edge you are tasked with evaluating fully before moving onto the next one.
i. It is helpful to quote supporting evidence from the provided texts and graphs before reasoning about their relevance to the final evaluation for that node or edge.
ii. While evaluating a node or edge, you may examine several plausible counterparts to judge presence, semantics, abstraction, etc. (e.g., to see if it is broader or narrower than any ground-truth items). Use all relevant comparisons to inform your decision, but output one—and only one—set of labels for the item.
b. Only focus on the specific type of evaluation you are asked to do. Regardless of the accuracy (or lack thereof) in other categories, if you are asked to evaluate node precision, only evaluate node precision, not recall or edges. These are intended to be separate evaluations, so do not conflate the two.
c. Not Applicable labels must be explicitly selected when a category is skipped due to prior labels
d. Be conservative when grading - When in doubt between two labels, ere on the side of being harsh.

Start by thinking step-by-step in `<think>` tags. Then, output your answer in a YAML code block, formatted exactly as specified in the expected output format.

LLM-as-a-Judge Prompt (2/7)

# Node Level Evaluation

## Node Precision
For each node in the LLM's response, evaluate against both ground truth sources:

1. Ground-Truth Graph Evaluation
- Explicitly identify and quote ALL potentially corresponding nodes from ground-truth graph
- Apply these labels where applicable:
Presence Labels (select one):
- PRESENCE_STRONG_MATCH: Core concept matches a ground-truth node with only minor, inconsequential differences
- PRESENCE_WEAK_MATCH: Core concept shares meaning with a ground-truth node, even if there are noticeable differences
- PRESENCE_NO_MATCH: There is no ground-truth node that captures a remotely similar core concept

Semantic Labels (select one):
- SEMANTIC_STRONG: Exactly or nearly identical meaning with only subtle distinctions
- SEMANTIC_MODERATE: Same core concept but with meaningful differences in scope or implication
- SEMANTIC_WEAK: Shares some semantic space but with substantial differences
- SEMANTIC_NA: Not applicable

Abstraction Labels (select one):
- ABSTRACTION_BROADER: Represents a more general concept that includes the ground-truth node
- ABSTRACTION_ALIGNED: Represents approximately the same scope and specificity of the ground-truth node
- ABSTRACTION_NARROWER: Represents a more specific subset of the ground-truth node
- ABSTRACTION_NA: Not applicable or the concepts were so different as to make abstraction comparison impossible

2. Ground-Truth Text Evaluation
- Explicitly quote ALL relevant supporting text from source
- Apply these labels where applicable:
Evidence Labels (select one):
- PRESENCE_STRONG_MATCH: Core concept appears in text with only minor, inconsequential differences
- PRESENCE_WEAK_MATCH: Core concept shares significant meaning with text but has notable differences
- PRESENCE_NO_MATCH: No text segments capture a similar core concept

LLM-as-a-Judge Prompt (3/7)

Semantic Labels (select one):
- SEMANTIC_STRONG: Captures precisely what is stated in text or represents meaning with minimal interpretation
- SEMANTIC_MODERATE: Requires some interpretation but maintains core meaning
- SEMANTIC_WEAK: Significant interpretation needed; meaning partially preserved
- SEMANTIC_NA: Not applicable

Abstraction Labels (select one):
- ABSTRACTION_BROADER: Represents a more general concept that includes text concepts
- ABSTRACTION_ALIGNED: Represents approximately the same scope and specificity as the text
- ABSTRACTION_NARROWER: Represents a more specific subset of text concepts
- ABSTRACTION_NA: Not applicable or the concepts were so different as to make abstraction comparison impossible

## Node Level Recall
For each node in the ground-truth graph, evaluate against the LLM's response:

Response Evaluation
- Explicitly identify and quote ALL potentially corresponding nodes from LLM's response
- Apply these labels where applicable:

Importance Labels (select one):
- IMPORTANCE_CORE: Ground-truth node represents a fundamental concept central to the causal structure
- IMPORTANCE_INTERMEDIATE: Ground-truth node serves as a key connection between central concepts
- IMPORTANCE_PERIPHERAL: Ground-truth node provides supplementary or contextual information

Presence Labels (select one):
- PRESENCE_STRONG_MATCH: Core concept appears in response with only minor, inconsequential differences
- PRESENCE_WEAK_MATCH: Core concept shares significant meaning with a response node but has notable differences
- PRESENCE_NO_MATCH: No response node captures a similar core concept

Semantic Labels (select one):
- SEMANTIC_COMPLETE: Ground-truth concept fully captured with high fidelity, whether in single or multiple nodes
- SEMANTIC_PARTIAL: Core aspects captured but with some meaning loss or missing implications
- SEMANTIC_MINIMAL: Only basic or surface-level aspects of the concept captured
- SEMANTIC_NA: Not applicable

LLM-as-a-Judge Prompt (4/7)

Abstraction Labels (select one):
- ABSTRACTION_BROADER: Represents a more general concept that includes the ground-truth node
- ABSTRACTION_ALIGNED: Represents approximately the same scope and specificity of the ground-truth node
- ABSTRACTION_NARROWER: Represents a more specific subset of the ground-truth node
- ABSTRACTION_NA: Not applicable or the concepts were so different as to make abstraction comparison impossible

# Edge Level Evaluation

## Edge Precision
For each edge (causal relationship) in the LLM's response, evaluate against both ground truth sources:

1. Ground-Truth Graph Evaluation
- Explicitly identify and quote ALL potentially corresponding edges from ground-truth graph
- Apply these labels where applicable:

Presence Labels (select one):
- PRESENCE_STRONG_MATCH: Edge connects highly similar concepts as in ground-truth
- PRESENCE_WEAK_MATCH: Edge connects somewhat similar concepts as in ground-truth
- PRESENCE_NO_MATCH: No corresponding edge exists in ground-truth

Directionality Labels:
- DIRECTION_CORRECT: Direction of causality matches ground-truth
- DIRECTION_REVERSED: Direction of causality is opposite of ground-truth
- DIRECTION_NA: Not applicable or the concepts were so different as to make direction comparison impossible

Abstraction Labels:
- ABSTRACTION_ALIGNED: Edge represents similar scope of relationship as ground-truth
- ABSTRACTION_BROADER: Edge is substantially more general than ground-truth
- ABSTRACTION_NARROWER: Edge is substantially more specific than ground-truth
- ABSTRACTION_NA: Not applicable or the concepts were so different as to make abstraction comparison impossible

LLM-as-a-Judge Prompt (5/7)

2. Ground-Truth Text Evaluation
- Explicitly quote ALL relevant supporting text that describes causal relationships
- Apply these labels where applicable:

Evidence Labels (select one):
- PRESENCE_GRAPH_ONLY: Causal relationship present in ground-truth graph (always select this if present)
- PRESENCE_EXPLICIT: Causal relationship directly stated in text (only if not in graph)
- PRESENCE_IMPLIED: Causal relationship can be reasonably inferred from text (only if not in graph)
- PRESENCE_NO_MATCH: No text supports this causal relationship (only if not in graph)

Inference Labels (select one):
- INFERENCE_DIRECT: Relationship matches text's explicit causal claims
- INFERENCE_DERIVED: Relationship logically follows from text
- INFERENCE_STRETCHED: Relationship possible but weakly supported
- INFERENCE_NA: Not applicable or relationship does not exist

Abstraction Labels (select one):
- ABSTRACTION_ALIGNED: Matches the granularity of text's causal claims
- ABSTRACTION_BROADER: Generalizes multiple textual relationships
- ABSTRACTION_NARROWER: Specifies a subset of text's causal claims
- ABSTRACTION_NA: Not applicable or the concepts were so different as to make abstraction comparison impossible

## Edge Level Recall
For each causal relationship (edge) in the ground-truth graph, evaluate against the LLM's response:

Response Evaluation
- Explicitly identify and quote ALL potentially corresponding causal relationships from LLM's response
- Apply these labels where applicable:

Importance Labels (select one):
Importance is based on how important it is to the ground-truth graph, regardless of whether it is present or accurately represented in the LLM's response.

- IMPORTANCE_CENTRAL: A key causal relationship that drives main effects
- IMPORTANCE_CONNECTING: Links major causal chains together
- IMPORTANCE_AUXILIARY: Provides supplementary causal context

LLM-as-a-Judge Prompt (6/7)

Presence Labels (select one):
- PRESENCE_STRONG_MATCH: Core concept appears in response with only minor, inconsequential differences
- PRESENCE_WEAK_MATCH: Core concept shares significant meaning with a response node but has notable differences
- PRESENCE_NO_MATCH: No response node captures a similar core concept

Directionality Labels (select one):
- DIRECTION_CORRECT: Causal relationship captured with correct direction
- DIRECTION_REVERSED: Causal relationship present but direction is reversed
- DIRECTION_UNCLEAR: Relationship present but direction is ambiguous
- DIRECTION_MISSING: Relationship entirely absent from response

Abstraction Labels (select one):
- ABSTRACTION_ALIGNED: One-to-one relationship match at similar level of detail
- ABSTRACTION_BROADER: Edge is substantially more general than ground-truth
- ABSTRACTION_NARROWER: Edge is substantially more specific than ground-truth
- ABSTRACTION_NA: Not applicable or the concepts were so different as to make abstraction comparison impossible

# Expected Output Format
The output should be in YAML format. Only include the evaluation sections that are being evaluated - omit other sections entirely. For example, if only evaluating node precision, only the node_precision_evaluations section should be present. However, within the required evaluation sections, be sure to always include the Not Applicable labels rather than omitting them.

LLM-as-a-Judge Prompt (7/7)

```
“‘yaml
# If evaluating node precision:
node_precision_evaluations:
- node_number: \verb¡integer¿——
graph_evaluation:
presence_label: <PRESENCE_LABEL>
semantic_label: <SEMANTIC_LABEL>
abstraction_label: <ABSTRACTION_LABEL>
text_evaluation:
presence_label: <PRESENCE_LABEL>
semantic_label: <SEMANTIC_LABEL>
abstraction_label: <ABSTRACTION_LABEL>

# If evaluating node recall:
node_recall_evaluations:
- node_number: <integer>
importance_label: <IMPORTANCE_LABEL>
presence_label: <PRESENCE_LABEL>
semantic_label: <SEMANTIC_LABEL>
abstraction_label: <ABSTRACTION_LABEL>

# If evaluating edge precision:
edge_precision_evaluations:
- edge_number: <integer>
graph_evaluation:
presence_label: <PRESENCE_LABEL>
directionality_label: <DIRECTION_LABEL>
abstraction_label: <ABSTRACTION_LABEL>
text_evaluation:
presence_label: <PRESENCE_LABEL>
inference_label: <INFERENCE_LABEL>
abstraction_label: <ABSTRACTION_LABEL>

# If evaluating edge recall:
edge_recall_evaluations:
- edge_number: <integer>
importance_label: <IMPORTANCE_LABEL>
presence_label: <PRESENCE_LABEL>
directionality_label: <DIRECTION_LABEL>

abstraction_label: <ABSTRACTION_LABEL>
“‘
```

LLM-as-a-Judge Scoring Mechanics

The quantitative metrics derived from the LLM-as-a-Judge's YAML output are calculated as follows. First, the judge's qualitative labels for various evaluation criteria (e.g., PRESENCE_STRONG_MATCH, SEMANTIC_MODERATE, IMPORTANCE_CORE) are mapped to pre-defined numerical scores, ranging from 0.0 (no match/irrelevant) to 1.0 (perfect match/highly important). For multi-faceted evaluations like node precision, which considers presence, semantic similarity, and abstraction level, a composite score for a single aspect (e.g., node precision against the ground-truth graph) is computed by averaging the numerical scores of its constituent labels.

Precision metrics (node precision, edge precision) for each item generated by the LLM are determined by comparing it against both the ground-truth graph and the source text. If the item is labeled as PRESENCE_NO_MATCH against both sources, its score is 0.0. Otherwise, the higher of the two composite scores (one from graph comparison, one from text comparison) is taken as the item's precision score. The overall precision for a category (e.g., node precision) is then the arithmetic mean of these individual item precision scores.

Recall metrics (node recall, edge recall) assess how well the LLM's output captures items from the ground-truth graph. For each ground-truth item, a composite correctness score is calculated based on its presence and the fidelity of its representation in the LLM's output (considering factors like semantics, abstraction, and directionality for edges). This correctness score is then multiplied by a numerical importance weight assigned by the judge to that ground-truth item (e.g., IMPORTANCE_CORE receives a higher weight than IMPORTANCE_AUXILIARY). The final recall score for a category is a weighted average: the sum of (correctness score × importance weight) for all ground-truth items, divided by the sum of all possible importance weights. This ensures that correctly recalling more important ground-truth items contributes more significantly to the recall score.

Finally, F1 scores for nodes, edges, and overall performance are calculated using the standard harmonic mean: $2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$. Overall precision and recall are micro-averaged, where the total weighted sum of correct predictions is divided by the total number of predictions (for precision) or total ground-truth items (for recall, considering importance weights), across both nodes and edges.

References

et al., D.-A. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv preprint arXiv:2501.12948.

Hosseinichimeh, N.; Majumdar, A.; Williams, R.; and Ghaffarzadegan, N. 2024. From text to map: a system dynamics bot for constructing causal loop diagrams. *System Dynamics Review*, 40(3): e1782.

Jin, Z.; Chen, Y.; Leeb, F.; Gresele, L.; Kamal, O.; Lyu, Z.; Blin, K.; Adauto, F. G.; Kleiman-Weiner, M.; Sachan, M.; and Schölkopf, B. 2024. CLadder: Assessing Causal Reasoning in Language Models. https://arxiv.org/abs/2312.04350.

Joshi, A.; Ahmad, A.; and Modi, A. 2024. COLD: Causal reasOning in cLosed Daily activities. arXiv:2411.19500.

Joshi, N.; Saparov, A.; Wang, Y.; and He, H. 2024. LLMs Are Prone to Fallacies in Causal Inference. arXiv:2406.12158.

Kim, H.; and Andersen, D. F. 2012. Building confidence in causal maps generated from purposive text data: mapping transcripts of the Federal Reserve. *System Dynamics Review*, 28(4): 311–328.

Kıcıman, E.; Ness, R.; Sharma, A.; and Tan, C. 2024. Causal Reasoning and Large Language Models: Opening a New Frontier for Causality. arXiv:2305.00050.

Lasheras, U. A.; and Pinheiro, V. 2025. CaLQuest.PT: Towards the Collection and Evaluation of Natural Causal Ladder Questions in Portuguese for AI Agents. In Hettiarachchi, H.; Ranasinghe, T.; Rayson, P.; Mitkov, R.; Gaber, M.; Premasiri, D.; Tan, F. A.; and Uyangodage, L., eds., *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, 325–343. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Miliani, M.; Auriemma, S.; Bondielli, A.; Chersoni, E.; Passaro, L.; Sucameli, I.; and Lenci, A. 2025. ExpliCa: Evaluating Explicit Causal Reasoning in Large Language Models. arXiv:2502.15487.

Oh, S. S. 2025. Plausibly Exogenous Galore. Notion. Available at https://sangmino.notion.site/1a897b8106ca44eeaf31dcd5ae5a61b1?v=ff7dc75862c6427eb4243e91836e077e.

OpenAI. 2025. OpenAI o3-mini System Card. OpenAI, January 31, 2025. https://cdn.openai.com/o3-mini-system-card.pdf.

Team, M. A. 2025. Mistral Small 3.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models!

Veldhuis, G. A.; Blok, D.; de Boer, M. H. T.; Kalkman, G. J.; Bakker, R. M.; and van Waas, R. P. M. 2024. From text to model: Leveraging natural language processing for system dynamics model development. System Dynamics Review, 40(3): e1780.

Wang, D.; Zuo, Y.; Li, F.; and Wu, J. 2024. LLMs as Zero-shot Graph Learners: Alignment of GNN Representations with LLM Token Embeddings. arXiv:2408.14512.

Yamin, K.; Gupta, S.; Ghosal, G. R.; Lipton, Z. C.; and Wilder, B. 2024. Failure Modes of LLMs for Causal Reasoning on Narratives. arXiv:2410.23884.

Zhou, Y.; Wu, X.; Huang, B.; Wu, J.; Feng, L.; and Tan, K. C. 2024. CausalBench: A Comprehensive Benchmark for Causal Learning Capability of LLMs. arXiv:2404.06349.