

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

Anonymous ACL submission

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 relies on synthetic or simplified texts with explicitly stated causal relationships. These texts typically feature short passages and few causal relations, failing to reflect the complexities of real-world reasoning. 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 (different levels of explicitness, number of causal events and relationships), and domain. 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 F_1 score of only 0.535. Through systematic analysis across aspects of real-world text (explicitness, number of causal events and relationships, length of text, domain), our benchmark offers targeted insights for further research into advancing LLM causal reasoning.

1 Introduction

The ability to identify and understand causal relationships embedded within texts is a fundamental aspect of human intelligence (Pearl, 2009; Gopnik et al., 2004) and is crucial for complex decision-making. Humans excel at inferring these relationships from text, even when they are not explicitly stated (Graesser et al., 1994). As a simplified example, given the text “Athletes spent extra time at the training facility this season. ... Coaches noticed improved attitudes during warmups. ...

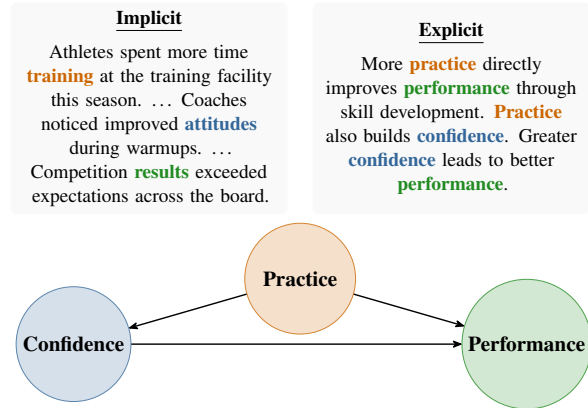


Figure 1: An example causal graph illustrating the difference between explicit and implicit texts describing the same causal relationships. The explicit text directly states causal relationships using clear language (“directly improves,” “leads to”). The implicit text describes the same relationships without stating causation (“spent more time training,” “noticed improved attitudes”). This exemplifies a key challenge with real-world texts, where causal reasoning must be used to construct the graph.

Competition results exceeded expectations across the board,”¹ humans identify the underlying causal events and infer the relationships: practice causes improved performance, practice causes increased confidence, and confidence causes better performance. Notably, none of these causal relationships need to be directly stated for a human to infer them.

Given the importance of causality to intelligence, significant research has focused on assessing the causal reasoning capabilities of large language models (LLMs). Much of this work evaluates LLMs’ stored knowledge about causal concepts (Zhou et al., 2024; Kıcıman et al., 2024; Miliani et al., 2025; Joshi et al., 2024a), rather than their ability to infer causal relationships from text. While some studies do examine extracting causal relationships from text, they often use synthetically generated or simplified texts with explicitly stated causal

¹Ellipses indicate omitted text between sentences, representing how real-world documents intersperse causal information throughout lengthy passages.

relations (Veldhuis et al., 2024; Hosseinichimeh et al., 2024; Oh, 2025; Jin et al., 2024; Joshi et al., 2024a; Lasheras and Pinheiro, 2025). This falls short of real-world scenarios where causal relationships are embedded in lengthy, complex texts with varying degrees of explicitness and diverse domains. Figure 1 illustrates this contrast: while synthetic texts may present clear, explicit causal statements, real-world texts convey causal relationships implicitly, requiring genuine causal reasoning to extract the underlying structure.

To address this gap, we introduce Real-world Causality from Textual Evidence (**ReCITE**), a novel benchmark drawn from real-world academic literature.² ReCITE features diverse texts varying in length, levels of explicitness, complexity (5–140 causal events, 6–205 relationships), and spanning multiple domains. To enable automated evaluation at scale, we develop an LLM-as-a-Judge framework for assessment of generated causal graphs. Our experiments on state-of-the-art (SOTA) LLMs using ReCITE demonstrate that LLMs struggle to infer causal relationships from real-world text, with the best models achieving an average F_1 score of only 0.535. 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 advancing LLM causal reasoning. Our main contributions are as follows:

- We propose ReCITE, the first benchmark to evaluate real-world causal reasoning abilities of LLMs from text, featuring samples with realistic variation in length, explicitness, causal complexity, and domain.
- We conduct extensive experiments on ten SOTA LLMs, revealing significant limitations in LLM’s ability to perform causal reasoning from real-world text, with the best-performing model achieving only 0.535 F_1 score. We also show that performance is poor even when identification of causal events is removed, indicating causal reasoning is the primary challenge.
- We develop a novel LLM-as-a-Judge evaluation framework that enables automated assess-

²Code and benchmark samples are included in supplementary materials. Full dataset to be released upon publication.

ment of generated graphs.

2 Related Work

Causal Reasoning with LLMs As LLMs have grown in capabilities, significant research has explored their causal reasoning abilities under varied evaluation styles (see surveys in (Kıcıman et al., 2024; Wan et al., 2025)). Some benchmarks focus on predicting causal structure from a provided variable set, largely bypassing long-form textual inference (e.g., CausalBench (Zhou et al., 2024)), while others utilize textual inputs but rely on synthetic or simplified narratives with explicit, often pairwise, relationships or small graphs (Chen et al., 2024; Jin et al., 2024; Miliani et al., 2025; Hosseinichimeh et al., 2024). A complementary line of work targets sentence-level causal language in real-world corpora (Ding et al., 2025) or evaluates causal understanding via question-answering over short contexts (Lasheras and Pinheiro, 2025; Chi et al., 2025; Joshi et al., 2024a). While these benchmarks are valuable, they fail to reflect real-world causal reasoning involving lengthy texts, large causal graphs, and varied levels of explicitness. Unlike prior benchmarks that rely on synthetic texts, short contexts, pairwise relations, or provided variable lists (see Appendix A), ReCITE requires full graph construction from lengthy real-world documents where causal relationships must be inferred rather than extracted from explicit statements.

Knowledge Graph Discovery with LLMs Recent work has applied LLMs to knowledge graph construction (Bratanić, 2024; Zhang and Soh, 2024; Yu et al., 2023), and related analyses in economics use language models to map concept-relationship graphs across large paper corpora (Garg and Fetzer, 2025). However, these approaches emphasize factual relations or coarse claim links, differing from the challenge in ReCITE of reconstructing the full causal graph from long-form text.

3 The ReCITE Benchmark

We present Real-world Causality from Textual Evidence (**ReCITE**), to the best of our knowledge, the first-ever realistic causal discovery from text benchmark. Given an input text, the task is to construct a causal graph where nodes represent causal events and directed edges represent their causal relationships. Constructing a realistic benchmark for this task requires source texts that exhibit real-world complexity and reliable ground-truth causal

graphs. Academic papers employing causal loop diagrams satisfy both requirements: they contain detailed textual descriptions of goals, methodology, background, and causal reasoning without relying on non-textual elements, and each paper centers around a single, human-authored causal graph that articulates its modeling assumptions. This enables accurate annotation of ground-truth graphs, and makes construction from the text feasible even when elements are not mentioned explicitly. We construct ReCITE through a systematic 3-step pipeline: (i) Academic paper collection, (ii) Causal Graph Annotation, (iii) Post-processing.

3.1 Academic Paper Collection

We utilize API access to academic paper repositories, restricting our search to open access sources (MDPI and PLOS) to ensure legal redistribution of texts and diagrams. We use a keyword search for the term “causal loop diagram,” resulting in 646 candidate papers. We then manually exclude all papers with graphs unsuitable for annotation, including ones with no causal graph 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 containing a single causal graph (either as the sole graph, or as the culmination of a sequence of graphs) so that it can serve as ground truth. During manual filtering, the location of each primary causal graph is labeled to ensure accurate annotation (e.g., “Top diagram on page 7”).

Table 1 displays summary statistics of the resulting corpus across key dimensions. Text lengths range from 684 to 171,141 characters. Explicitness varies substantially, with 87.7% of causal events mentioned in the text. The corpus also spans 17 distinct fields: external classification via OpenAlex Topics (Priem et al., 2022) reveals that only 5.5% of papers fall under core Economics, Econometrics & Finance, with Environmental Science (20.2%), Engineering (20.2%), and Business/Management (12.0%) comprising the largest shares. Embedding analysis confirms greater similarity to STEM literature than traditional economics (MMD = 0.141 vs. 0.114; Appendix F). Semantic diversity is similarly high: across 6,378 unique causal concepts, 92.9% appear in only a single paper, and 97.6% of paper pairs share zero causal concepts. Full graph topology statistics, including density, cyclicity, and motif distributions, are reported in Appendix E.

Attribute	Mean \pm SD	Range
Number of Samples	–	292
Text Length	40541 \pm 17722	(684, 171141)
Nodes per Graph	25.0 \pm 15.8	(5, 140)
Edges per Graph	37.4 \pm 24.3	(6, 205)
Explicitness per Node	0.877	–

Table 1: ReCITE dataset statistics showing its real-world attributes across key dimensions. Text length is measured in characters. Explicitness per Node reports the proportion of nodes mentioned in the sample’s text.

3.2 Causal Graph Annotation

As ReCITE 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, consistent with prior work (White and Barbrook-Johnson, 2025; Bai et al., 2025). Therefore, we employ annotators to convert each causal graph into a standardized format (“source_variable” \rightarrow “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.

3.3 Post-Processing

Graph Post-Processing To ensure the reliability of ground-truth graphs and mitigate potential human error, we utilize a rigorous post-processing pipeline. 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 automatically corrected are flagged and manually corrected.

To verify transcription quality, a second annotator re-labeled 37 randomly chosen diagrams (879 edges, 674 nodes; \approx 15% of ReCITE). We observed 22 missing and 5 spurious edges, giving edge-level precision = 0.994, recall = 0.975, $F_1 = 0.984$, and Cohen’s $\kappa = 0.987$. 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 are provided in Appendix I. We explore utilization of code-based approaches for correction of node naming, but find that it is prone to erroneously combining distinct nodes (e.g., GDP and GNP). To address this, we utilize LLMs for automated correction (see Appendix S).

Text Post-Processing To convert PDF papers into a textual format suitable for LLM input, which cannot be done trivially with automated parsing (Meuschke et al., 2023), we utilize a multi-step pipeline. First, 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, due to the infeasibility of code-based approaches. Manual testing shows current LLMs cannot perform this process accurately in one step. 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.

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

Model	Type	Parameters
Claude Opus 4.5 (Anthropic, 2025)	Reasoning	N/A
GPT-5.2 (OpenAI, 2025b)	Reasoning	N/A
Gemini 3 Pro (Google DeepMind, 2025)	Reasoning	N/A
Gemini 3 Flash (Google DeepMind, 2025)	Reasoning	N/A
GLM 4.7 [†] (Zhipu AI, 2025)	Reasoning	355
Kimi K2 [†] (Moonshot AI, 2025)	Reasoning	1000
DeepSeek R1 [†] (et al., 2025)	Reasoning	671
QwQ 32B [†] (Team, 2025b)	Reasoning	32
Qwen 2.5 32B [†] (Team, 2024)	Instruct	32
Llama 3.1 8B [†] (Grattafiori et al., 2024)	Instruct	8

Table 2: Evaluated models. [†]Open-weight. Type denotes reasoning-tuned versus instruction-tuned. Parameters listed in billions, and N/A when not disclosed.

4 Experiments

In this section, we examine the performance of SOTA LLMs on ReCITE. 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 ReCITE, including text length, explicitness, number of nodes and edges, and domain? Finally, we demonstrate common reasoning failures via a case study.

4.1 Experiment Setup

Models. We evaluate diverse SOTA models spanning reasoning and instruction-tuned categories, proprietary and open-weight, as summarized in Table 2. We assess in the zero-shot setting and use default hyperparameters. We exclude samples where the model’s maximum context length is exceeded (only Qwen 2.5 32B is effected, with 20 samples excluded), and applies across all experiments except where otherwise specified. For proprietary models, we retry responses where we fail to receive an answer. We explicitly provide the expected number of nodes because the task is otherwise ill-defined; there are many valid levels of abstraction for each graph, and real-world tasks typically specify the desired granularity. We enforce strict JSON formatting for graph outputs to enable automated evaluation. For responses that fail formatting requirements, we apply a post-processing step using Mistral Small (Team, 2025a) or GPT-5-mini (OpenAI, 2025a) to convert malformed outputs into valid JSON while preserving the intended relationships. We use DeepSeek v3.2 (DeepSeek-AI et al., 2025) as our judge, due to its long context, tractable cost, and is not an evaluated model.

Evaluation Metrics. Evaluating causal graph construction from text poses challenges that tra-

Model	Node Precision (\uparrow)	Node Recall (\uparrow)	Edge Precision (\uparrow)	Edge Recall (\uparrow)	F ₁ (\uparrow)	SHD (\downarrow)	Normalized SHD (\downarrow)
Claude Opus 4.5	0.972 \pm 0.055	0.571 \pm 0.215	0.853 \pm 0.154	0.305 \pm 0.208	0.535 \pm 0.202	47.0 \pm 36.5	0.108 \pm 0.091
GPT 5.2	0.960 \pm 0.070	0.524 \pm 0.214	0.804 \pm 0.160	0.309 \pm 0.205	0.510 \pm 0.201	58.5 \pm 45.1	0.130 \pm 0.094
Gemini 3 Pro	0.980 \pm 0.050	0.527 \pm 0.232	0.865 \pm 0.152	0.281 \pm 0.215	0.501 \pm 0.221	48.3 \pm 37.2	0.114 \pm 0.101
GLM 4.7	0.970 \pm 0.085	0.524 \pm 0.226	0.808 \pm 0.169	0.274 \pm 0.199	0.493 \pm 0.210	50.8 \pm 37.8	0.116 \pm 0.089
Kimi K2	0.950 \pm 0.079	0.508 \pm 0.221	0.778 \pm 0.177	0.253 \pm 0.189	0.471 \pm 0.197	54.7 \pm 44.6	0.126 \pm 0.100
Gemini 3 Flash	0.962 \pm 0.096	0.501 \pm 0.234	0.818 \pm 0.164	0.255 \pm 0.210	0.469 \pm 0.218	50.2 \pm 36.4	0.119 \pm 0.098
DeepSeek R1	0.955 \pm 0.086	0.504 \pm 0.204	0.773 \pm 0.181	0.231 \pm 0.182	0.460 \pm 0.186	50.9 \pm 36.0	0.122 \pm 0.100
QwQ 32B	0.944 \pm 0.088	0.481 \pm 0.209	0.730 \pm 0.221	0.209 \pm 0.167	0.432 \pm 0.183	51.0 \pm 37.1	0.117 \pm 0.090
Qwen 2.5 32B	0.925 \pm 0.109	0.431 \pm 0.219	0.686 \pm 0.222	0.180 \pm 0.161	0.386 \pm 0.192	49.9 \pm 31.6	0.107 \pm 0.085
Llama 3.1 8B	0.916 \pm 0.156	0.341 \pm 0.205	0.615 \pm 0.256	0.111 \pm 0.130	0.295 \pm 0.171	51.8 \pm 30.9	0.103 \pm 0.080

Table 3: Comparison of different models’ performance on ReCITE (mean \pm standard deviation).

ditional metrics cannot address. First, unlike traditional causal discovery, where nodes are provided, this task requires assessing both node identification and edge prediction. Second, a generated element may be valid if supported by either the ground-truth graph or the source text; traditional metrics only compare against ground-truth graphs. Standard causal discovery metrics (precision, recall, F₁, Structural Hamming Distance (SHD) (Tsamardinos et al., 2006)) are therefore insufficient.

To address these challenges, we design an LLM-as-a-Judge framework (Zheng et al., 2023). The framework evaluates at both node and edge levels, and compares generated elements against both the ground-truth graph and source text. Because real-world texts use synonyms and varying levels of abstraction, predicted and ground-truth elements often differ in surface form while representing the same underlying concept. The framework therefore, assesses semantic similarity and abstraction levels to enable meaningful comparison. The judge processes each evaluation dimension (node precision, node recall, edge precision, edge recall) in separate prompts; combining all evaluations into a single prompt degrades accuracy due to the complexity of jointly assessing lengthy texts, generated graphs, and ground-truth graphs. We use this framework to compute three metrics:

Precision measures whether each generated element is valid. Each predicted node or edge is compared against both the ground-truth graph and the source text along three dimensions: (1) *presence*—whether the concept appears in the reference (strong match, weak match, or no match); (2) *semantic similarity*—how closely the meaning aligns (strong, moderate, weak, or N/A); and (3) *abstraction level*—whether it is broader, aligned, or narrower than the reference. If an element receives “no match” against both ground-truth graph and source text, it scores 0.0; otherwise, the higher of the two composite scores is used.

Recall measures whether the generated graph cap-

tures ground-truth elements. Not all causal relationships are equally central to a graph’s structure; missing a primary causal driver is more severe than missing a peripheral modifier. We therefore assess each ground-truth element for its *importance* (core, intermediate, or peripheral) and how well it is captured by the generated graph. Recall is computed as a weighted average where correctness scores are multiplied by importance weights.

SHD is computed from the judge’s edge-level evaluations as the sum of three error types: false positives (predicted edges with no ground-truth match), false negatives (ground-truth edges not captured), and reversals (edges with correct nodes but inverted direction). Normalized SHD divides by $n \times (n - 1)$, where n is the number of nodes, yielding a value in $[0, 1]$. Comprehensive rubric and scoring mechanics are in Appendices Q and Z.

We validate judge reliability through cross-judge agreement analysis across four candidate judges (DeepSeek R1, DeepSeek v3.2, Gemini 3 Flash, GLM 4.7), finding strong pairwise correlations ($r = 0.78\text{--}0.91$) that indicate consistent rankings regardless of judge choice. We further confirm no family bias: DeepSeek v3.2 scores DeepSeek R1 outputs no more favorably than outputs from other model families (Appendix C).

4.2 Performance of State-of-the-Art LLMs

To assess the performance of LLMs on causal discovery from real-world text, we benchmark them on ReCITE. As shown in Table 3, all models perform poorly, with the best-performing model, Claude Opus 4.5, achieving just an average F₁ of 0.535 across all samples. This shows even SOTA LLMs struggle at causal reasoning from real-world text. We observe that reasoning models showed the best performance, and that there is 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. Fine-grained error analysis reveals that direction reversals are rare ($< 1.1\%$ of matched edges), indicating that when models identify correct relationships, they almost never invert causality. However, while 85–90% of generated edges have some textual support, only 17–33% match ground-truth edges, suggesting models generate plausible causal relationships from the text but fail to recover the specific relationships encoded in the ground-truth graph (Appendix O). Due to the realistic nature of the benchmark, there is a high standard deviation across metrics, with difficulty varying based on factors including explicitness, number of nodes, number of edges, text size, and domains mimicking the diverse conditions for real-world causal reasoning.

Model	Precision (\uparrow)	Recall (\uparrow)	F ₁ (\uparrow)	SHD (\downarrow)
Claude Opus 4.5	0.582	0.536	0.551	35.6
Gemini 3 Pro	0.567	0.531	0.538	35.8
Gemini 3 Flash	0.552	0.526	0.528	37.8
GPT 5.2	0.537	0.521	0.520	42.6
Kimi K2	0.515	0.513	0.505	45.2
GLM 4.7	0.517	0.514	0.505	41.0
DeepSeek R1	0.513	0.512	0.502	40.6
QwQ 32B	0.483	0.501	0.481	41.0
Qwen 2.5 32B	0.290	0.435	0.332	51.0
Llama 3.1 8B	0.218	0.409	0.266	62.9

Table 4: Name-assisted performance where models are provided ground-truth node names. Performance remains poor despite removing node identification, showing that causal reasoning is the primary challenge.

Node Identification vs. Causal Reasoning. To investigate whether poor model performance stems from node identification or causal reasoning, and to validate our LLM-as-a-Judge framework, we conduct an ablation experiment in which models are explicitly provided with the complete set of ground-truth node names. This enables deterministic evaluation, allowing us to compare rankings across evaluation conditions.

Table 4 shows results from this controlled scenario. While models trivially achieved perfect node-level precision and recall by design, improvements in causal inference were limited: edge-level F₁ slightly improved for stronger models such as R1 (by only +0.042) and Gemini 3 Pro (by

+0.037). Meanwhile, weaker models like Qwen 2.5 32B and Llama 3.1 8B saw reductions in performance, declining by -0.054 and -0.029 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. This ablation highlights that the poor performance of LLMs on ReCITE is due to fundamental limitations with causal reasoning, not entity recognition. The consistency of model rankings across deterministic and LLM-as-a-Judge evaluations provides empirical validation of our evaluation framework: systematic judge biases would manifest as ranking instability between settings, which we do not observe. We further validate robustness by computing unweighted metrics using binary matching without importance weights; model rankings remain highly consistent (Spearman’s $\rho = 0.915$, Kendall’s $\tau = 0.778$; Appendix D).

4.3 Performance Analysis

In this section, we analyze how factors contributing to real-world complexity (explicitness, text length, number of nodes, number of edges, and domain) affect LLM causal reasoning performance.

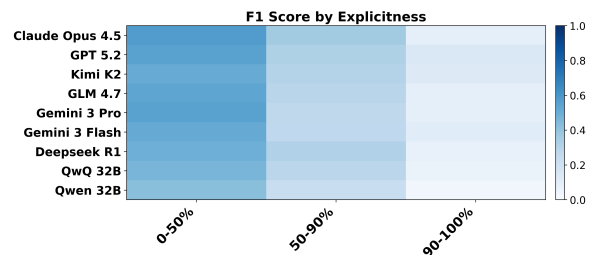


Figure 2: Heatmap of the average model scores across bins, where 100% is entirely explicit, and F₁ score from 0-1, showcasing how explicitness has a large impact on performance, and LLMs struggle to infer causality when explicit references are sparse.

Explicitness A key challenge in real-world causal reasoning is that causal events are not always explicitly stated in the text. To analyze how this affects performance, we measure the explicitness of each sample: the degree to which causal events in the ground-truth graph are mentioned in the source text. We develop an explicitness score that categorizes each node into three levels: (1)

Model	Text	Edges	Nodes
Gemini 3 Pro	+0.08 ± 0.22	-0.03 ± 0.22	-0.02 ± 0.22
Claude Opus 4.5	+0.02 ± 0.19	-0.02 ± 0.21	-0.00 ± 0.19
GLM 4.7	+0.02 ± 0.20	-0.06 ± 0.21	-0.04 ± 0.20
Kimi K2	+0.01 ± 0.19	-0.03 ± 0.20	-0.01 ± 0.20
GPT 5.2	-0.01 ± 0.20	-0.03 ± 0.21	-0.02 ± 0.20
Gemini 3 Flash	-0.01 ± 0.22	-0.05 ± 0.22	-0.04 ± 0.21
DeepSeek R1	-0.01 ± 0.17	-0.04 ± 0.19	-0.02 ± 0.18
Qwen 2.5 32B	-0.02 ± 0.19	-0.12 ± 0.20	-0.12 ± 0.19
QwQ 32B	-0.02 ± 0.18	-0.03 ± 0.19	-0.03 ± 0.19
Llama 3.1 8B	-0.03 ± 0.17	-0.08 ± 0.18	-0.07 ± 0.17

Table 5: 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.

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 (see Appendix R), then calculate the explicitness as:

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

where E represents explicitly mentioned nodes, I represents implicitly mentioned nodes, and A is the nodes that are absent from mention. 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 explicitness score provides a quantitative way to assess how much causal reasoning (versus simple text comprehension) is required for each sample.

Samples with high degrees of explicitness 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 explicitness. As shown in Figure 2, there is a strong positive correlation between a sample’s explicitness and model performance (reported using F_1). Additionally, all models struggle even for the most explicit samples, showing that LLMs struggle with the task even under easier conditions. This performance further degrades for the least explicit texts, with F_1 dropping by around half for all models between the samples with lowest and highest levels of explicitness. We verify that this relationship holds under a stricter definition of explicitness in Appendix K.

Effect of Size. As ReCITE 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 5. Counterintuitively, there is a weak, but positive correlation between size and performance. Given the strong effect of explicitness on performance shown in Section 4.3, we investigate the relationship between explicitness 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, positive correlation with level of explicitness (with R^2 values of 0.171, 0.002, and 0.002 respectively). We find weak but significant negative correlations between graph complexity metrics (edges, motif counts) and model performance ($|r| = 0.13\text{--}0.18$, $p < 0.05$). However, when we bin samples by graph density or cyclicity, performance remains largely flat across both dimensions. Notably, the 90.4% of graphs containing feedback cycles show nearly identical performance to directed acyclic graphs (Appendix E).

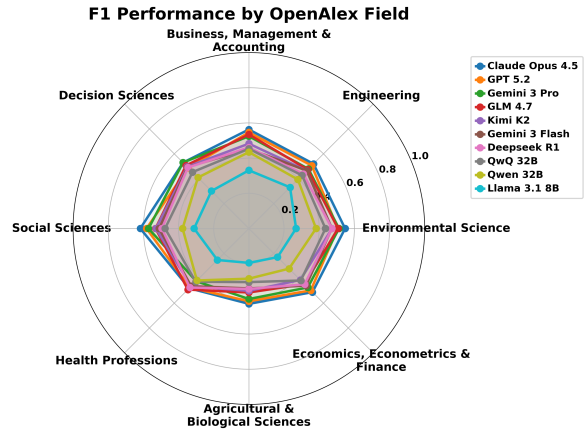


Figure 3: Radar chart of domain-specific accuracies for each model, depicting the accuracy of different models.

Effect of Domain and Data Diversity. To assess whether semantic domain affects performance, we cluster paper embeddings using k-means ($k=4$, selected by silhouette score) and analyze F_1 by cluster. HDBSCAN found no natural clusters, classifying all papers as a noise distribution across semantic space rather than topical concentration. Cluster differences are not statistically significant (ANOVA $F = 1.03$, $p = 0.38$, $\eta^2 = 0.012$; Kruskal-Wallis $H = 3.42$, $p = 0.33$), with a performance gap of only 0.041 F_1 between best and worst clusters. As shown in Figure 3, the best-performing model overall, Claude Opus 4.5, outperforms on 6 out of

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 4: Verbatim excerpts from R1's reasoning trace. Ellipses added for readability and bold for emphasis.

8 domains, with Gemini 3.0 Pro having a higher score in Decision Sciences and GLM 4.7 in Health Professions. However, all models perform poorly regardless of domain, indicating that domain does not substantially impact difficulty. We also analyze semantic diversity via embedding-based comparisons against reference corpora and test whether semantic domain predicts task difficulty through cluster-based performance analysis; results are reported in Appendix F.

4.4 Case Study

To further analyze ReCITE, we select a benchmark sample about livelihood efficiency in the Qinba Mountains and the graph constructed by 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.”

The ground-truth graph captures these relationships directly: *Annual precipitation*, *Annual sunshine hours*, and *Arable land in county* each cause *Total grain output*. Despite this being explicitly described in the source text, R1 fails to faithfully

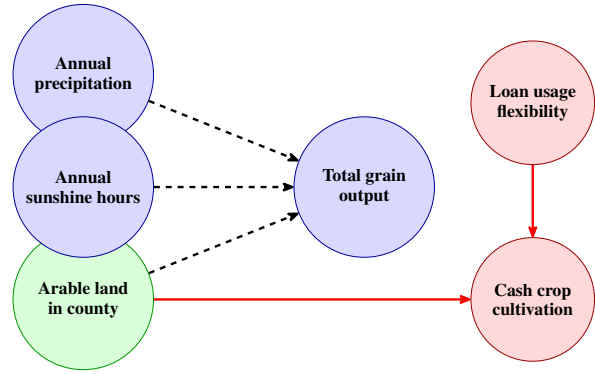


Figure 5: Causal subgraph annotated with R1's performance. Green: concepts R1 correctly identified. Blue: ground-truth nodes not included. Dashed: ground-truth edges not included. Red: spurious nodes and edges.

reconstruct the graph. As shown in Figure 4, R1 initially identifies precipitation as a relevant factor during reasoning, but omits both *Annual precipitation* and *Annual sunshine hours* from its final graph. While R1 correctly identifies *Arable land in county*, it connects this node to *Cash crop cultivation* rather than *Total grain output*. R1 also introduces spurious structure, adding *Loan usage flexibility* linked to *Cash crop cultivation*, despite loan flexibility relating to financial capital rather than crop selection. As shown in Figure 5, R1 recovers only 1 of 4 ground-truth nodes, misses all 3 ground-truth edges, and adds 2 spurious nodes with 2 spurious edges. This illustrates how LLMs struggle to integrate causally relevant information dispersed across lengthy real-world documents, even when individual cues are straightforward. We include a complementary human expert case study in Appendix M, demonstrating that the task is tractable for domain experts.

5 Conclusions

In this paper, we introduce ReCITE, the first benchmark to assess causal reasoning from text under realistic conditions. ReCITE draws diverse samples from academic literature, featuring texts varied in length, graph complexity, and domain. Experiments show that SOTA LLMs struggle significantly, with the best model achieving just 0.535 F_1 . Performance degrades further as causal relationships become less explicit, and remains poor even when node names are provided, indicating the primary challenge is causal reasoning itself, not entity recognition. ReCITE provides a cost-effective platform (all experiments under \$1000) for future research into these fundamental limitations.

619 **Limitations**

620 We acknowledge the following limitations: (i) Our
621 evaluation methodology incorporates LLMs for
622 post-processing and judging, which may introduce
623 model-specific biases and calibration differences.
624 We validate reliability through cross-judge agree-
625 ment analysis (showing strong pairwise correla-
626 tions, $r > 0.78$) and verify absence of family bias,
627 though LLM judgments may not perfectly align
628 with expert human preferences. (ii) Models are pro-
629 vided with the target number of nodes, simplifying
630 the task compared to fully unconstrained causal dis-
631 covery; this constraint is necessary to disentangle
632 causal reasoning ability from granularity prefer-
633 ences and to enable consistent evaluation across
634 samples with varying complexity. (iii) Our corpus
635 is drawn from open-access academic literature, and
636 while we demonstrate diversity across domains and
637 graph structures, performance on ReCITE may not
638 fully generalize to other text genres or domains not
639 represented in the benchmark. (iv) Our benchmark
640 is English-only; while the underlying causal rea-
641 soning task is language-agnostic, performance on
642 ReCITE may not generalize to other languages. (v)
643 We evaluate models exclusively in the zero-shot set-
644 ting with default hyperparameters. While all mod-
645 els have been trained to utilize chain-of-thought
646 (which they were prompted to do), techniques like
647 few-shot prompting or retrieval-augmented gen-
648 eration may further improve performance. (vi)
649 ReCITE draws from academic papers employing
650 causal loop diagrams, a methodology from the field
651 of system dynamics. Model performance on this
652 benchmark may not generalize to causal graphs
653 constructed under different paradigms.

654 **Ethics Statement**

655 ReCITE is constructed from open-access academic
656 papers published under CC BY 4.0 licenses (MDPI
657 and PLOS), which permit redistribution and deriva-
658 tive works. We do not anticipate significant risk
659 of personally identifiable information or offensive
660 content, as the source materials are academic pub-
661 lications; we did not perform additional screen-
662 ing for such content beyond the peer review al-
663 ready applied by the source venues. Annotation
664 was performed by undergraduate economics stu-
665 dents, whose participation was optional, and were
666 informed that their annotations would be used in an
667 academic publication. Further details provided in
668 Appendix I. ReCITE is intended for research eval-

uation of LLMs to better understand their causal
reasoning capabilities. We acknowledge there is al-
ways a risk of misuse by bad actors, but believe this
risk is outweighed by furthering the understanding
of the capabilities of these models. AI assistants
were heavily used in this work for the data pro-
cessing pipeline, as detailed in Section 3. They
were also used to aid with writing code, literature
review, and minor revisions to the paper; however,
all AI-related changes were manually reviewed and
approved by the authors.

References 680

- Anthropic. 2025. [Claude opus 4.5: Model card](#). Techni-
cal report. 681
682
- Nikolay Babakov, Ehud Reiter, and Alberto Bugarín-
Diz. 2025. [CausalGraphBench: a benchmark for
evaluating language models capabilities of causal
graph discovery](#). In *Proceedings of the 63rd Annual
Meeting of the Association for Computational Lin-
guistics (Volume 4: Student Research Workshop)*,
pages 240–258, Vienna, Austria. Association for
Computational Linguistics. 683
684
685
686
687
688
689
690
- Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He,
Zongbo Han, Zheng Zhang, and Mike Zheng Shou.
2025. [Hallucination of multimodal large language
models: A survey](#). *Preprint*, arXiv:2404.18930. 691
692
693
694
- T. Bratanić. 2024. Building knowledge graphs with
llm graph transformer: A deep dive into langchain’s
implementation of graph construction with llms. To-
wards Data Science. 695
696
697
698
- Sirui Chen, Bo Peng, Meiqi Chen, Ruiqi Wang, Mengy-
ing Xu, Xingyu Zeng, Rui Zhao, Shengjie Zhao,
Yu Qiao, and Chaochao Lu. 2024. [Causal evaluation
of language models](#). *Preprint*, arXiv:2405.00622. 699
700
701
702
- Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan,
Xiaoguang Ren, Tongliang Liu, and Bo Han. 2025.
[Unveiling causal reasoning in large language models:
Reality or mirage?](#) *Preprint*, arXiv:2506.21215. 703
704
705
706
- DeepSeek-AI, A. Liu, Aoxue Mei, Bangcai Lin, Bing
Xue, Bingxuan Wang, and ... 2025. [Deepseek-v3.2:
Pushing the frontier of open large language models](#).
arXiv preprint. ArXiv:2512.02556. 707
708
709
710
- Xiaohan Ding, Kaike Ping, Buse Çarık, and Eugenia
Rho. 2025. [A multi-level benchmark for causal
language understanding in social media discourse](#).
In *Proceedings of the 2025 Conference on Empiri-
cal Methods in Natural Language Processing*, pages
28764–28778, Suzhou, China. Association for Com-
putational Linguistics. 711
712
713
714
715
716
717
- DeepSeek-AI et al. 2025. [DeepSeek-R1: Incentiviz-
ing reasoning capability in llms via reinforcement
learning](#). arXiv preprint arXiv:2501.12948. 718
719
720

721	Prashant Garg and Thiemo Fetzer. 2025. Causal claims in economics . <i>Preprint</i> , arXiv:2501.06873.	773
722		774
723	Google DeepMind. 2025. Gemini 3: Technical report . Technical report.	775
724		776
725	Alison Gopnik, Clark Glymour, David M. Sobel, Laura E. Schulz, Tamar Kushnir, and David Danks. 2004. A theory of causal learning in children: Causal maps and bayes nets . <i>Psychological Review</i> , 111(1):3–32.	777
726		778
727		779
728		780
729		781
730	Arthur C. Graesser, Murray Singer, and Tom Trabasso. 1994. Constructing inferences during narrative text comprehension . <i>Psychological Review</i> , 101(3):371–395.	782
731		783
732		784
733		785
734	Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models . <i>Preprint</i> , arXiv:2407.21783.	786
735		787
736		788
737		789
738		790
739		791
740		792
741		793
742	Niyousha Hosseinichimeh, Aritra Majumdar, Ross Williams, and Navid Ghaffarzadegan. 2024. From text to map: a system dynamics bot for constructing causal loop diagrams . <i>System Dynamics Review</i> , 40(3):e1782.	794
743		795
744		796
745		797
746		798
747	Z. Jin, Y. Chen, F. Leeb, L. Gresele, O. Kamal, Z. Lyu, K. Blin, F. G. Adatao, M. Kleiman-Weiner, M. Sachan, and B. Schölkopf. 2024. CLadder: Assessing causal reasoning in language models . https://arxiv.org/abs/2312.04350 .	799
748		800
749		801
750		802
751		803
752	Abhinav Joshi, Areeb Ahmad, and Ashutosh Modi. 2024a. Cold: Causal reasoning in closed daily activities . <i>Preprint</i> , arXiv:2411.19500.	804
753		805
754		806
755	Nitish Joshi, Abulhair Saparov, Yixin Wang, and He He. 2024b. LLMs are prone to fallacies in causal inference . <i>Preprint</i> , arXiv:2406.12158.	807
756		808
757		809
758	Hyunjung Kim and David F. Andersen. 2012. Building confidence in causal maps generated from purposive text data: mapping transcripts of the federal reserve . <i>System Dynamics Review</i> , 28(4):311–328.	810
759		811
760		812
761		813
762	Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2024. Causal reasoning and large language models: Opening a new frontier for causality . <i>Preprint</i> , arXiv:2305.00050.	814
763		815
764		816
765		817
766	Uriel Anderson Lasheras and Vladia Pinheiro. 2025. CaLQuest.PT: Towards the collection and evaluation of natural causal ladder questions in Portuguese for AI agents . In <i>Proceedings of the First Workshop on Language Models for Low-Resource Languages</i> , pages 325–343, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.	818
767		819
768		820
769		821
770		822
771		823
772		824
	Donggyu Lee, Sungwon Park, Yerin Hwang, Hyoshin Kim, Hyunwoo Oh, Jungwon Kim, Meeyoung Cha, Sangyoon Park, and Jihee Kim. 2025. Benchmarking llm causal reasoning with scientifically validated relationships . <i>Preprint</i> , arXiv:2510.07231.	
	Norman Meuschke, Apurva Jagdale, Timo Spinde, Jelena Mitrović, and Bela Gipp. 2023. A Benchmark of PDF Information Extraction Tools Using a Multi-task and Multi-domain Evaluation Framework for Academic Documents , page 383–405. Springer Nature Switzerland.	
	Martina Miliani, Serena Auriemma, Alessandro Bondielli, Emmanuele Chersoni, Lucia Passaro, Irene Sucameli, and Alessandro Lenci. 2025. Explica: Evaluating explicit causal reasoning in large language models . <i>Preprint</i> , arXiv:2502.15487.	
	Moonshot AI. 2025. Kimi k2: Reasoning model card . Model card.	
	Sangmin S. Oh. 2025. Plausibly exogenous galore . Notion. Available at https://sangmino.notion.site/1a897b8106ca44eeaf31dcd5ae5a61b1?v=ff7dc75862c6427eb4243e91836e077e .	
	OpenAI. 2025a. Gpt-5 system card . Official model documentation and system card.	
	OpenAI. 2025b. Gpt-5.2 system card . System card.	
	OpenAI. 2025. Openai o3-mini system card . OpenAI, January 31, 2025. https://cdn.openai.com/o3-mini-system-card.pdf .	
	Judea Pearl. 2009. <i>Causality</i> . Cambridge University Press, Cambridge.	
	Jason Priem, Heather Piwowar, and Richard Orr. 2022. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts . <i>Preprint</i> , arXiv:2205.01833.	
	Mistral AI Team. 2025a. Mistral small 3 .	
	Qwen Team. 2024. Qwen2.5: A party of foundation models!	
	Qwen Team. 2025b. Qwq-32b: Embracing the power of reinforcement learning .	
	Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. 2006. The max-min hill-climbing bayesian network structure learning algorithm . <i>Machine Learning</i> , 65:31–78.	
	G. A. Veldhuis, D. Blok, M. H. T. de Boer, G. J. Kalkman, R. M. Bakker, and R. P. M. van Waas. 2024. From text to model: Leveraging natural language processing for system dynamics model development . <i>System Dynamics Review</i> , 40(3): e1780.	
	Guangya Wan, Yunsheng Lu, Yuqi Wu, Mengxuan Hu, and Sheng Li. 2025. Large language models for causal discovery: Current landscape and future directions . <i>Preprint</i> , arXiv:2402.11068.	

825	Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024.	comprehension, to be used. Lastly, we also exam-	876
826	Llms as zero-shot graph learners: Alignment of gnn	ine the realism (of the source data) - is it drawn	877
827	representations with llm token embeddings . <i>Preprint</i> ,	from the real-world? The following table offers a	878
828	arXiv:2408.14512 .	high-level visual comparison, with further details	879
829	Jordan White and Pete Barbrook-Johnson. 2025. Using	on each benchmark discussed subsequently.	880
830	vision-language models to extract network data from		
831	images of system maps . <i>INET Oxford Working Paper</i>	A.1 Discussions of Selected Benchmarks	881
832	<i>Series</i> , (2025-26).		
833	Khurram Yamin, Shantanu Gupta, Gaurav R. Ghosal,	The following discussions provide context for the	882
834	Zachary C. Lipton, and Bryan Wilder. 2024. Failure	data presented in Table 6, highlighting their ap-	883
835	modes of llms for causal reasoning on narratives .	proaches and how we notably differ from this prior	884
836	<i>Preprint</i> , arXiv:2410.23884 .	work. While these works are valuable, prior work	885
837	S. Yu, T. Huang, M. Liu, and Z. Wang. 2023. BEAR:	fails to measure the causal reasoning abilities of	886
838	Revolutionizing service domain knowledge graph	LLMs from text under real-world settings. We	887
839	construction with llm . In Monti, F., Rinderle-Ma, S.,	highlight the approach of each benchmark, and	888
840	Ruiz Cortés, A., Zheng, Z., and Mecella, M. (Eds.),	how they compare to ReCITE.	889
841	<i>Service-Oriented Computing. ICSOC 2023. Lecture</i>	Several benchmarks concentrate on pairwise	890
842	<i>Notes in Computer Science</i> , vol 14419, pp. 339–346.	causal relations, include inputs that are highly ex-	891
843	Springer, Cham.	PLICIT, or have inputs that are generated syntheti-	892
844	B. Zhang and H. Soh. 2024. Extract, define, canonical-	cally or are crafted as short texts by hand. We dif-	893
845	ize: An llm-based framework for knowledge graph	fer from these approaches by aiming to construct	894
846	construction. <i>arXiv preprint arXiv:2404.03868</i> .	large graphs from real-world literature. ExpliCa	895
847	Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan	(Miliani et al., 2025) examines how LLMs un-	896
848	Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin,	derstand explicit connectives in sentence pairs, re-	897
849	Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang,	sulting in 2-node links. By design, it uses very	898
850	Joseph E. Gonzalez, and Ion Stoica. 2023. Judg-	short, often crafted inputs and focuses on explicit	899
851	ing llm-as-a-judge with mt-bench and chatbot arena .	cues, thereby avoiding the complexities of implicit	900
852	<i>Preprint</i> , arXiv:2306.05685 .	causality and information integration from exten-	901
853	Zhipu AI. 2025. Glm-4.7: General language model	sive texts that ReCITE targets. Meanwhile, LLM	902
854	technical report . Technical report.	Fallacies (Joshi et al., 2024b) employs short, syn-	903
855	Yu Zhou, Xingyu Wu, Beicheng Huang, Jibin Wu, Liang	thetic scenarios to test pairwise causal inference,	904
856	Feng, and Kay Chen Tan. 2024. Causalbench: A	focusing on logical fallacies when LLMs are pre-	905
857	comprehensive benchmark for causal learning capa-	sented with explicit non-causal information. While	906
858	bility of llms . <i>Preprint</i> , arXiv:2404.06349 .	this addresses a specific type of reasoning com-	907
859	A Detailed Comparison with Causal	plexity, its synthetic and brief inputs differ greatly	908
860	Reasoning Benchmarks	from the real-world, extensive texts and broader	909
861	This section provides a comparative overview of	graph construction task in ReCITE. Plausibly Ex-	910
862	ReCITE against existing benchmarks for causal	ogenous Galore (Oh, 2025) serves as an interest-	911
863	reasoning. While each benchmark contributes to	ing bridge, as it uses long economics documents	912
864	understanding LLM capabilities, ReCITE is specif-	similar to ReCITE. However, its task is to find only	913
865	ically designed to test the construction of complex	the main pairwise link for the entire document,	914
866	causal graphs from long-form, real-world academic	which greatly limits its diversity and complexity.	915
867	texts. We evaluate benchmarks based on their di-	Another group of benchmarks attempt graph con-	916
868	versity and complexity. In this context, we consider	struction, but typically rely on short, simplified,	917
869	a benchmark diverse when it spans many domains	or synthetic descriptions. These lack the depth	918
870	or sub-domains, or draws data from many different	and realism of long, real-world texts. From text	919
871	types of sources. Meanwhile, we deem a sample	to map (Hosseinichimeh et al., 2024) generate	920
872	complex when there are varied degrees of explicit-	relatively small graphs (max 9-15 nodes) from	921
873	ness in samples; that is, that the benchmark features	concise, hand-crafted descriptions. Such inputs	922
874	many samples that are not highly explicit, requir-	inherently limit textual diversity and likely fea-	923
875	ing that causal reasoning, rather than mere reading	ture more explicit causal links, sidestepping the	924
		challenge of parsing lengthy, nuanced documents	925
		with varying levels of explicitness. Failure Modes	926

Benchmark	Primary Task	Long Text	Input Type	Max Nodes	Diverse	Complex	Realism
ExpliCa (Miliari et al., 2025)	Pairwise ID	✗	Sentences	2	✗	✗	✗
LLM Fallacies (Joshi et al., 2024b)	Pairwise Inference	✗	Scenario/Vignette	2	✗	⚠	✗
Plausibly Exogenous (Oh, 2025)	Pairwise ID	✓	Full Document	2	✓	⚠	✓
Lee et al. (Lee et al., 2025)	Question About Triplet	✗	Triplet/Question	3	✓	✗	✗
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 (Kicman et al., 2024)	Multiple Tasks	✗	Mixed	Varies	⚠	⚠	⚠
CausalTalk (Ding et al., 2025)	Multi-task Causal Language	✗	Social Media Posts	N/A	✓	⚠	✓
CaLM (Chen et al., 2024)	Multiple Tasks	✗	Mixed	20	⚠	✗	✗
CausalProbe-2024 (Chi et al., 2025)	Causal QA	✗	QA/Short Context	N/A	✗	⚠	⚠
CLadder (Jin et al., 2024)	Causal QA	✗	Narrative/Question	4	✗	✓	✗
CausalGraphBench (Babakov et al., 2025)	Causal Structure ID	✗	Variable List	222	⚠	⚠	✗
CausalBench (Zhou et al., 2024)	Causal Structure ID	✗	Query/Question	109	⚠	⚠	✓
COLD (Joshi et al., 2024a)	Causal QA	✗	Query/Question	33	⚠	✓	✓
CaLQuest.PT (Lasheras and Pinheiro, 2025)	Causal QA	✗	Query/Question	N/A	✓	✓	✓
ReCITE (Ours)	Graph Construction	✓	Full Document	140	✓	✓	✓

Table 6: Visual comparison of ReCITE 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 ReCITE’s focus on long-text graph extraction.

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 ReCITE, and its narratives are purpose-built rather than reflecting the reasoning of real-world conditions. Lee et al. (Lee et al., 2025) extracts causal triplets from titles and abstracts of literature, and asks questions about that ground-truth triplet. However, unlike ReCITE, this relies on the model’s pre-existing knowledge as no source text is provided as grounding, and tests only for the ability to answer binary questions rather than open-ended causality. Similar to ReCITE, **CausalGraphBench** (Babakov et al., 2025) derives its samples from published causal graphs; however, its task is not causal reasoning from text, requiring only determining the connections between the provided list of nodes given a short synthetic description (e.g., 78 word description for a graph with over 750 edges).

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 (Kicman 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 ReCITE’s reliance on extensive, unmodified academic texts for end-to-end graph extraction. **CausalTalk** (Ding et al., 2025) is another form of sentence-level analysis, where evaluated models are instructed to extract the shortest phrase that describes the explicit or implicit causal relationship from a social media post. While this uses real-world text, it uses small excerpts and does not require the model to do open-ended reasoning, thus avoiding many of the complexities of real-world texts where identifying causality is not as simple as quoting it directly from the document.

To the best of our knowledge, ReCITE is the first benchmark for LLMs to measure the causal reasoning abilities from long, diverse, and complex, real-world texts. While previous benchmarking efforts have explored lengthy, diverse, and complex texts, causal reasoning, and real-world conditions separately, we are the first to do so at once.

B Effect of Knowledge Cutoff

To investigate whether performance on ReCITE is influenced by pre-training contamination, we conduct an analysis comparing performance for each model before and after their knowledge cutoff. We select three models with varying cutoff dates: Llama 3.1 8B (December 2023), R1 (July 2024),

and Qwen 2.5 32B (September 2024). Our dataset contains 35 samples derived from papers published in 2024 or later. For Llama 3.1 8B, the average F_1 score increased from 0.288 on samples from before its knowledge cutoff to 0.339 on newer samples; for R1, performance increased slightly from 0.459 to 0.476; while Qwen 2.5 32B showed a decrease from 0.388 to 0.342. Despite this variation in performance, none are statistically significant. This suggests that models’ ability to identify causal relationships is not related to training on these specific documents during pre-training.

C Inter-Judge Agreement and Bias Analysis

To validate our LLM-as-a-Judge evaluation methodology, we measured agreement between multiple judge models and analyzed potential biases.

C.1 Overall Inter-Judge Agreement

Table 7 reports pairwise correlations between four candidate judges on overlapping evaluation subsets.

Judge 1	Judge 2	n	r	Mean Δ	Std Δ
DeepSeek R1	DeepSeek v3.2	959	0.820	+0.004	0.120
DeepSeek v3.2	Gemini 3 Flash	595	0.895	-0.122	0.090
DeepSeek R1	Gemini 3 Flash	463	0.782	-0.116	0.134

Table 7: Inter-judge agreement on large evaluation subsets. All correlations are strong ($r > 0.78$, $p < 10^{-30}$), indicating consistent rankings. Mean Δ shows differences (positive denotes first judge scores higher).

All judge pairs show strong correlations ($r = 0.78$ – 0.90), indicating that all LLM judges produce consistent rankings regardless of differences in raw scores. This high level of agreement lends credibility to the effectiveness of our judging method.

C.2 Cross-Judge Analysis

To further validate consistency, we had all four candidate judges (DeepSeek R1, DeepSeek v3.2, Gemini 3 Flash, and GLM 4.7) evaluate the same 41 randomly selected DeepSeek R1 outputs. Table 8 shows all pairwise correlations on this matched subset.

C.3 Family Bias Analysis

A potential concern with LLM judges is self-bias. While our judge, DeepSeek v3.2 is not an evaluated model, it is in the DeepSeek family of models,

Judge 1	Judge 2	r	Mean Δ
DeepSeek R1	DeepSeek v3.2	0.863	-0.055
DeepSeek R1	Gemini 3 Flash	0.912	+0.070
DeepSeek R1	GLM 4.7	0.788	-0.020
DeepSeek v3.2	Gemini 3 Flash	0.863	-0.055
DeepSeek v3.2	GLM 4.7	0.912	+0.070
Gemini 3 Flash	GLM 4.7	0.912	+0.070

Table 8: Pairwise judge agreement on 41 matched samples evaluated by all four judges. There is a strong correlations ($r = 0.79$ – 0.91) across all pairs, validating that inter-judge agreement is not specific to a particular judge.

and thereby could score DeepSeek R1 more favorably. To test this, we compare the calibration gap between v3.2 and Gemini 3 Flash when judging outputs from different model families.

Outputs From	n	v3.2 – Gemini	DeepSeek?
DeepSeek R1	292	-0.128	Yes
Llama 3.1 8B	214	-0.114	No
QwQ 32B	89	-0.122	No

Table 9: Calibration gap between DeepSeek v3.2 and Gemini 3 Flash by output source. Negative values indicate v3.2 scores higher. The gap is consistent (-0.11 to -0.13) regardless of whether the outputs come from a DeepSeek model, suggesting no detectable family bias.

DeepSeek v3.2 scores outputs approximately 0.12 points higher than Gemini 3 Flash across all model families. The difference between judging DeepSeek R1 outputs (-0.128) versus non-DeepSeek outputs (-0.114 to -0.122) is minimal (0.006 – 0.014) is well within the noise of calibration differences. This suggests that v3.2 does not give preferential treatment to DeepSeek outputs.

C.4 Judge Selection

We select DeepSeek v3.2 as our primary judge because it is not among the evaluated models, shows strong performance at long context lengths, shows high agreement with other judges ($r > 0.82$), and its computational costs are tractable.

D Unweighted Evaluation Metrics

We also report unweighted version of our fine-grained evaluation, acting as an alternative measure of performance. This metric uses binary matching (any match versus no match), removes importance weighting, partial credit for the wrong level of abstraction, and does not provide credit when the gen-

erated graph includes elements that are supported by the text but not the graph. Specifically, for unweighted evaluation, we derive binary decisions from the LLM judge’s labels:

- **Match:** PRESENCE_STRONG_MATCH or PRESENCE_WEAK_MATCH
- **No Match:** PRESENCE_NO_MATCH

Precision and recall are computed against the ground-truth graph only (with no credit for grounding solely from the text), with all elements weighted equally:

$$\text{Precision} = \frac{|\text{Predicted} \cap \text{Ground-Truth}|}{|\text{Predicted}|} \quad (1)$$

$$\text{Recall} = \frac{|\text{Predicted} \cap \text{Ground-Truth}|}{|\text{Ground-Truth}|} \quad (2)$$

Tables 10 and Table 11 present the unweighted metrics. Under this stricter evaluation, edge F_1 scores range from 0.111 to 0.297, substantially lower than the fine-grained scores in Table 3. Additionally, the gap between node and edge performance is even more pronounced under unweighted evaluation. The best-performing model achieves 0.592 node F_1 but only 0.297 edge F_1 , reinforcing our finding that causal edge inference, rather than node identification, is the primary bottleneck. Model rankings remain largely consistent with the fine-grained evaluation, with Claude Opus 4.5 leading across all metrics and Llama 3.1 8B trailing. Notably, GPT 5.2 exhibits the highest edge recall (0.290) but lower precision (0.258), suggesting it generates more edges overall at the cost of accuracy.

Rank Agreement. To quantify the consistency between weighted and unweighted evaluation, we compute rank correlation statistics across overall F_1 scores. The two methods show strong agreement (Spearman’s $\rho = 0.915$, $p < 0.001$; Kendall’s $\tau = 0.778$, $p < 0.001$). The top-performing model (Claude Opus 4.5) and bottom three models (QwQ 32B, Qwen 2.5 32B, Llama 3.1 8B) maintain identical rankings across both metrics. Minor rank changes occur in the middle of the distribution: GPT 5.2 drops two positions under unweighted evaluation due to its recall-heavy strategy receiving less benefit, while DeepSeek R1 gains two positions. This strong rank correlation validates that our fine-grained scoring preserves relative model performance while providing more nuanced assessment of partial matches and abstraction differences.

Model	Prec.	Recall	F_1
Claude Opus 4.5	0.350	0.283	0.297
Gemini 3 Pro	0.317	0.266	0.275
GLM 4.7	0.304	0.252	0.260
GPT 5.2	0.258	0.290	0.253
Gemini 3 Flash	0.282	0.239	0.239
Kimi K2	0.267	0.237	0.229
DeepSeek R1	0.277	0.217	0.226
QwQ 32B	0.272	0.201	0.213
Qwen 2.5 32B	0.228	0.170	0.173
Llama 3.1 8B	0.188	0.105	0.111

Table 10: Unweighted edge-level metrics from the LLM Judge using binary matching against the ground-truth graph.

Model	Prec.	Recall	F_1
Claude Opus 4.5	0.603	0.600	0.592
Gemini 3 Pro	0.568	0.557	0.551
DeepSeek R1	0.568	0.553	0.545
GPT 5.2	0.551	0.557	0.543
GLM 4.7	0.560	0.551	0.543
QwQ 32B	0.550	0.522	0.522
Kimi K2	0.530	0.543	0.518
Gemini 3 Flash	0.519	0.534	0.514
Qwen 2.5 32B	0.566	0.476	0.489
Llama 3.1 8B	0.508	0.384	0.402

Table 11: Unweighted node-level metrics from the LLM Judge using binary matching against the ground-truth graph.

E Graph Structural Diversity

We compute graph topology metrics across all 292 benchmark samples to characterize structural diversity. Table 12 reports summary statistics and Figure 6 visualizes distributions.

Key Observations. Node counts range from 5 to 140 and edge counts from 6 to 205. Most graphs are sparse (median density 0.070). 90.4% of graphs contain at least one cycle, with 63.6% of nodes participating in cycles on average. Chains ($A \rightarrow B \rightarrow C$) are the most prevalent motif, followed by colliders ($A \rightarrow C \leftarrow B$) and forks ($A \rightarrow B, A \rightarrow C$). All motif types show high variance across samples.

Across all 7,297 nodes: mean degree is 2.97, 18.8% have degree 1, 31.1% have degree 2, and 50.1% have degree 3+.

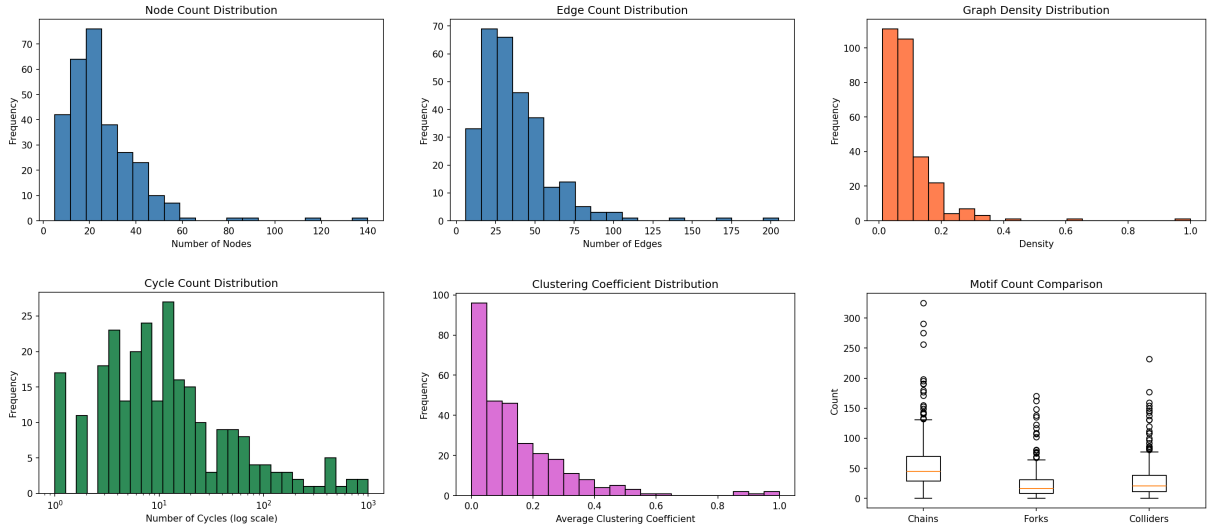


Figure 6: Distribution of graph topology metrics. Top row: node count, edge count, density. Bottom row: cycle count (log scale), clustering coefficient, motif counts.

Metric	Mean	Std	Min	Median	Max
Nodes	25.0	15.8	5	22.0	140
Edges	37.0	23.9	6	32.5	205
Density	0.094	0.089	0.011	0.070	1.000
Avg Degree	3.0	0.8	1	2.9	10
Sources	4.8	6.6	0	3.0	40
Sinks	1.8	4.2	0	0.0	42
Clustering	0.146	0.167	0.000	0.106	1.000
Cycles	50.7	245.2	0	8.0	3820
% in Cycles	63.6	31.7	0.0	72.7	100.0
SCCs	11.9	13.9	1	7.0	92
Largest SCC	13.4	9.4	1	12.0	42
Path Length	3.49	1.20	1.00	3.41	7.32
Betweenness Var	0.014	0.014	0.000	0.011	0.089
Chains	57.7	47.3	0	45.0	325
Forks	24.1	26.8	0	16.0	170
Colliders	31.5	32.8	0	21.0	232

Table 12: Graph topology statistics (N=292). SCC = Strongly Connected Component.

E.1 Topology-Performance Correlation

We compute correlations between topology metrics and model F_1 scores across 290 samples with complete evaluation data (Table 13).

Edge count and all three motif types show weak but significant negative correlations with F_1 ($|r| = 0.13\text{--}0.18$), indicating that larger, more complex graphs are somewhat harder. However, these effect sizes are small compared to the impact of textual explicitness reported in Section 4.3, where F_1 drops by approximately half between the most and least explicit samples. Graph structure contributes to difficulty but is not the primary bottleneck.

Metric	Pearson r	p	Spearman ρ	p
Nodes	-0.131*	0.026	-0.086	0.143
Edges	-0.175*	0.003	-0.133*	0.023
Density	+0.088	0.135	+0.048	0.418
Avg Degree	+0.003	0.959	-0.058	0.322
Sources	-0.029	0.619	-0.074	0.210
Sinks	-0.011	0.850	-0.029	0.620
Clustering	-0.035	0.556	-0.093	0.113
Cycles	-0.060	0.306	-0.074	0.209
% in Cycles	+0.001	0.986	+0.014	0.814
SCCs	-0.062	0.293	-0.046	0.432
Largest SCC	-0.100	0.087	-0.081	0.168
Betweenness Var	+0.036	0.543	+0.059	0.315
Chains	-0.176*	0.003	-0.139*	0.018
Forks	-0.154*	0.009	-0.140*	0.017
Colliders	-0.169*	0.004	-0.125*	0.032

Table 13: Topology- F_1 correlations (N=292). * indicates $p < 0.05$.

E.2 Density and Cyclicity Analysis

To further investigate whether graph structure affects difficulty, we bin samples by density and cyclicity. Table 14 shows performance across density bins; Table 15 compares acyclic (DAG) versus cyclic graphs. Performance is largely flat across both dimensions, with no consistent trends. Notably, the 90.4% of graphs containing feedback cycles show nearly identical performance to DAGs, indicating that cyclicity does not pose an additional reasoning challenge for LLMs. These results reinforce our finding that graph topology contributes minimally to task difficulty compared to textual explicitness.

Model	Sparse (<0.15)	Moderate (0.15–0.3)	Dense (≥0.3)
Claude Opus 4.5	0.538	0.520	0.514
GPT 5.2	0.512	0.498	0.513
Gemini 3 Pro	0.506	0.493	0.420
GLM 4.7	0.492	0.503	0.484
Kimi K2	0.477	0.437	0.437
Gemini 3 Flash	0.468	0.482	0.449
DeepSeek R1	0.460	0.452	0.491
QwQ 32B	0.427	0.455	0.477
Qwen 2.5 32B	0.375	0.452	0.406
Llama 3.1 8B	0.291	0.313	0.331

Table 14: F_1 score by graph density. Sample sizes: sparse (n=248), moderate (n=34), dense (n=10).

Model	Acyclic (DAG)	Cyclic
Claude Opus 4.5	0.534	0.535
GPT 5.2	0.518	0.509
Gemini 3 Pro	0.511	0.500
DeepSeek R1	0.503	0.455
GLM 4.7	0.503	0.491
Kimi K2	0.489	0.469
QwQ 32B	0.467	0.429
Gemini 3 Flash	0.462	0.470
Qwen 2.5 32B	0.406	0.384
Llama 3.1 8B	0.312	0.293

Table 15: F_1 score by cyclicity. Sample sizes: acyclic (n=28), cyclic (n=264).

F Textual Diversity Analysis

We validate the domain diversity of ReCITE using external bibliometric classification (OpenAlex Topics) and embedding-based analysis.

F.1 Domain Classification via OpenAlex

We map all 292 benchmark papers to OpenAlex Works (Priem et al., 2022) using DOIs and extract their primary topic classifications. Table 16 shows the domain distribution: less than 40% of papers fall under Social Sciences (where economics resides), with the plurality (47%) classified as Physical Sciences. At the field level (Table 17), only 5.5% are classified under core Economics, Econometrics & Finance. The benchmark spans 17 distinct fields, with Environmental Science and Engineering each comprising 20%.

Domain-level Shannon entropy is 1.12 (effective number of domains: 3.06/4). Field-level entropy is

Domain	Count	%
Physical Sciences	138	47.3
Social Sciences	108	37.0
Health Sciences	28	9.6
Life Sciences	18	6.2

Table 16: Primary domain distribution via OpenAlex Topics.

Field	Count	%
Environmental Science	59	20.2
Engineering	59	20.2
Business, Management & Accounting	35	12.0
Decision Sciences	28	9.6
Social Sciences	25	8.6
Health Professions	19	6.5
Agricultural & Biological Sciences	18	6.2
Economics, Econometrics & Finance	16	5.5
Other (9 fields)	33	11.3

Table 17: Primary field distribution across 17 fields.

2.30 (effective number: 9.93/17). This high entropy indicates broad coverage rather than concentration.

F.2 Embedding Analysis

We embed all paper abstracts using GTE-large-en-v1.5 and compare against reference corpora sampled from OpenAlex (Priem et al., 2022): 250 economics papers and 248 non-economics papers (physics, biology, computer science, medicine). Figure 7 shows the UMAP projection.

Maximum Mean Discrepancy between distributions shows ReCITE is more similar to STEM literature (MMD=0.141) than to economics (MMD=0.114). Mean pairwise cosine distance within ReCITE (0.419) slightly exceeds that of the economics reference (0.414).

F.3 Cluster-Based Performance Analysis

To test whether semantic similarity predicts task difficulty, we cluster ReCITE embeddings using k-means (k=4, selected by silhouette score) and analyze F_1 by cluster. HDBSCAN found no natural clusters, classifying all papers as noise. This indicates that there is broad distribution across semantic space.

Cluster differences are not statistically significant (ANOVA $F = 1.03$, $p = 0.38$; Kruskal-Wallis $H = 3.42$, $p = 0.33$; $\eta^2 = 0.012$). The

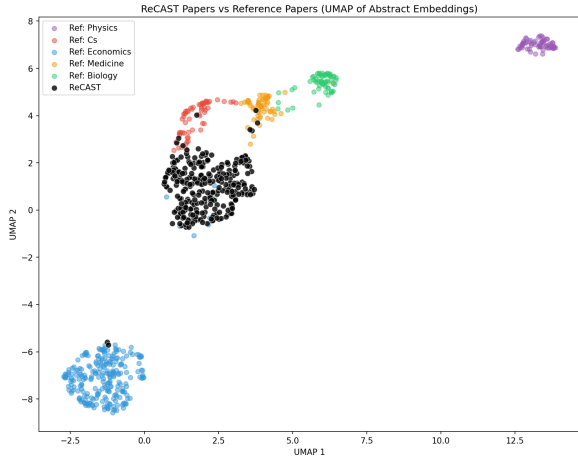


Figure 7: UMAP projection of ReCITE papers (black) against reference corpora. ReCITE papers are largely separate from traditional economics (bottom-left), reflecting their interdisciplinary nature.

Cluster	N	Mean F_1	Std
Engineering	72	0.413	0.146
Business/Management	89	0.411	0.170
Health Professions	54	0.399	0.147
Environmental Science	77	0.372	0.153

Table 18: Performance by embedding cluster.

performance gap between best and worst clusters is only 0.041 F_1 . Combined with the null effect of graph topology (Appendix E), this indicates that neither semantic domain nor structural complexity predicts difficulty, but rather textual explicitness remains the dominant factor.

G 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 9 and Figure 10. 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.

Model	Median Length	r	p -value
Kimi K2	29K	0.168	0.004*
DeepSeek R1	22K	0.163	0.005*
Claude Opus 4.5	15K	0.096	0.101
QwQ 32B	24K	0.093	0.113
GLM 4.7	28K	0.049	0.408
Llama 3.1 8B	3K	-0.030	0.699

Table 19: Reasoning trace length statistics and correlation with F_1 score. Median length measured in characters. Only Kimi K2 and DeepSeek R1 show statistically significant correlations ($*p < 0.05$) between reasoning length and performance, though effect sizes are small ($r < 0.2$). Llama 3.1 8B, a non-reasoning model, produces substantially shorter traces (3K vs. 15-29K for reasoning models).

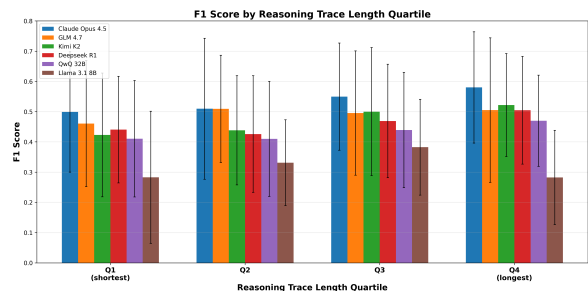


Figure 8: F_1 score by reasoning trace length quartile, computed per-model. Most models show modest performance gains with longer reasoning traces, with Claude Opus 4.5 improving from 0.50 (Q1) to 0.58 (Q4). However, Llama 3.1 8B shows no improvement across quartiles, and large error bars indicate substantial within-quartile variance for all models. This suggests that while extended reasoning provides some benefit, it is not a reliable predictor of performance.

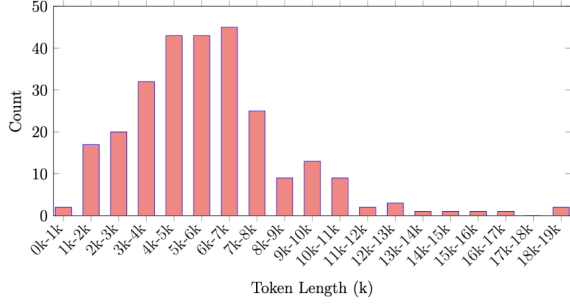


Figure 9: QwQ token length distribution. The data suggests that QwQ maintains a relatively consistent distribution across token lengths, reflecting its structured handling of reasoning chains. However, performance did not scale proportionally with longer traces, indicating limitations in handling extended reasoning efficiently.

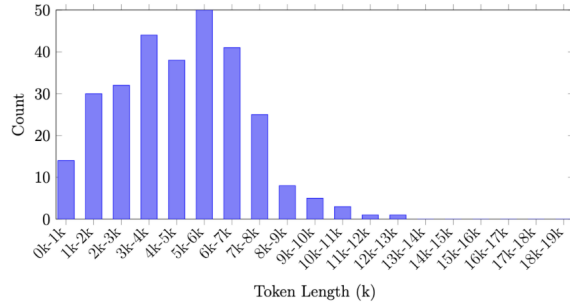


Figure 10: Distribution of reasoning trace lengths (in tokens) for R1 across benchmark samples. The model tends to produce mid-length reasoning traces (2,000–7,000 tokens), with very few exceeding 12,000 tokens.

H Size vs. Explicitness

We provide the following charts as a reference to visualize the relationship between size and explicitness.

I 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 ($\sim 13\%$ of the full dataset). These diagrams included 879 directed edges and 674 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 879 annotated edges, 22 edges were missed by the second an-

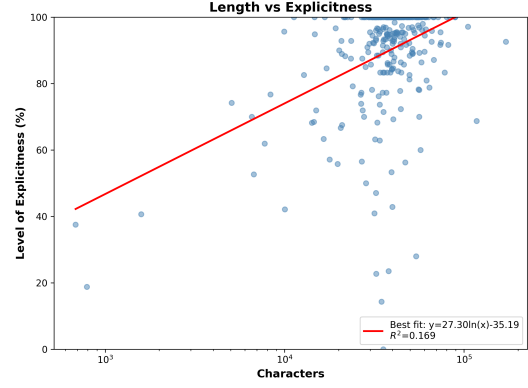


Figure 11: **Relationship between text length and explicitness under the strict setting.** This scatter plot shows each sample’s character count on the x-axis versus its level of explicitness on the y-axis. There is a modest positive correlation ($R^2 = 0.169$), indicating that longer texts tend to be more explicit. This helps explain why models perform slightly better on larger input instances, as increased explicitness makes causal edges easier to identify.

notator (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:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{857}{857 + 5} = 0.994,$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{857}{857 + 22} = 0.975,$$

$$F_1 = 0.984,$$

$$\text{SHD} = 27,$$

$$\kappa = 0.99$$

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 674 nodes, we observed:

- 8 auto-correctable typos (e.g., “carbondioxide” \rightarrow “carbon dioxide”).
- 4 minor differences (e.g., prefix/suffix omissions such as “CO₂ emissions” vs. “emissions”).

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	0.993 ± 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 20: 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.

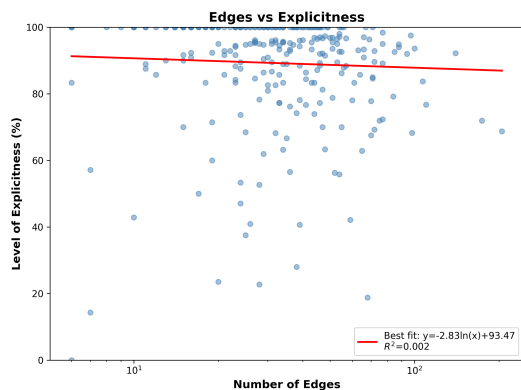


Figure 12: **Relationship between edges and explicitness.** This chart plots the count of true causal edges for each sample against its explicitness. With a very small positive correlation ($R^2 = 0.002$), edge density has minimal impact on how many confounders remain implicit. Models thus face similar challenges inferring hidden variables regardless of edge complexity.

- No major label mismatches (0 spurious or missing nodes).

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

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.

Additional Annotation Details. We utilize 20 undergraduate economics students as annotators. It was provided as an optional opportunity for extra credit, with credit proportional to the number of nodes labeled. Annotators would submit using the annotator form showed in Figure 13. Annotators were also explicitly informed that the annotation work would be utilized and published in an academic work. Formal written consent was not

collected, as this study was determined to be exempt from IRB review given the use of published academic materials. We believe the standard educational incentives of extra credit acted as adequate “payment” for their work, which we estimate took roughly 1 hour per graph (though this greatly varies based on the size and complexity of the graph). After all graphs were completed, extra credit was assigned, and all records of annotator names were permanently destroyed to preserve privacy.

1268
1269
1270
1271
1272
1273
1274
1275
1276
1277

Figure 13: **Screenshot of the annotator form.**

When manually excluding causal graphs, we exclude workshop papers, as these texts do not include sufficient details about the graph to make it identifiable. Most were able to be automatically removed by removing any papers whose abstracts include the keywords “workshop” or “group model build”, with the rest excluded via manual review.

1278
1279
1280
1281
1282
1283
1284

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 21: 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.

Model	Std.	Name	Δ
GPT 5.2	0.927	0.928	+0.001
Claude Opus 4.5	0.926	0.923	-0.003
GLM 4.7	0.926	0.924	-0.002
Kimi K2	0.926	0.926	+0.000
Gemini 3 Pro	0.925	0.923	-0.002
DeepSeek R1	0.925	0.924	-0.001
Gemini 3 Flash	0.924	0.922	-0.002
QwQ 32B	0.922	0.924	+0.002
Qwen 2.5 32B	0.913	0.918	+0.005
Llama 3.1 8B	0.909	0.908	-0.002

Table 22: Cosine similarity ($\Delta = \text{Name} - \text{Std.}$) under standard vs name-assisted TEA-GLM conditions. Minimal differences indicate that Name-Assist does not significantly enhance alignment with ground truth.

J 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 22), 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 22 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 +0.005 (for Qwen 2.5 32B), and several models, such as (Claude Opus 4.5) even decrease by -0.003 . 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.

K Alternative Measure of Explicitness

As discussed in the main paper, explicitness has a noticeable effect on model performance. For this, we determine whether a node is explicitness by determining if it is directly mentioned in the text (or not). Given the large effect of level of explicitness in performance, we explore if this relationship holds under different measures of the explicitness. As detailed in Section R, 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 explicitness, where we measure whether a node was mentioned explicitly, or if it was implicit/absent. We recalculate the explicitness for each sample for the “strict” measure as shown below.

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

Figure 14: Level of $\text{DC}_{\text{strict}}$ (explicitness) 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 explicitness score and compute the average over all nodes in the sample.

Under this alternative measure of explicitness, at a given level of explicitness, 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 15, the negative relationship between explicitness and performance holds. Performance is also consistently worse for all models under the

strict explicitness definition rather than the lenient one, as expected, adding credibility to the validity of the automated labeling.

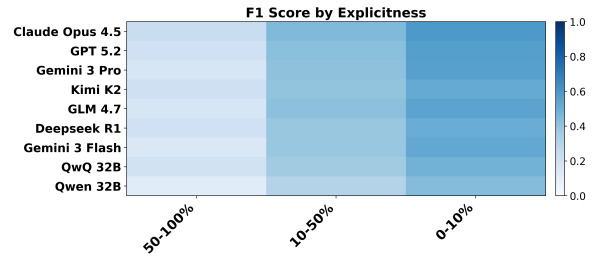


Figure 15: Average model scores across bins, where 0% is entirely explicit and 100% is completely not explicit (strict definition).

L Computational Costs

Despite the large size of ReCITE samples, its execution is notably quite computationally efficient. The total monetary cost for all experiments, encompassing the evaluation of all ten LLMs across the main task and all ablation studies, including multiple LLM-as-a-Judge evaluations, remained under \$1000. 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 ReCITE can be utilized and extended by researchers without imposing prohibitive computational or financial burdens.

M 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 com-

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"}
  ]
}
```
```

lowing 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.

1442

## O Fine-Grained Error Analysis

1443

We analyze the distribution of judge labels across edge precision evaluations to characterize model failure modes. Table 23 reports key label distributions. We find direction reversals are rare: when models identify edges matching the ground truth, they almost never invert causality (<1.1% across all models). Second, there is a large gap between text-grounding and graph-matching, with roughly 85–90% of generated edges have some textual support (EXPLICIT, IMPLIED, or GRAPH\_ONLY), yet only 17–33% match ground-truth edges. This indicates that models generate plausible causal relationships from the text but fail to recover the specific relationships in the ground-truth graph. Third, abstraction levels are typically appropriate: when edges match, ALIGNED dominates over BROADER or NARROWER. These patterns reinforce our finding that the primary bottleneck is

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461

identifying the correct causal relationships, not directionality or abstraction.

| Model           | Graph Match | Direction Correct | Direction Reversed | Text Support |
|-----------------|-------------|-------------------|--------------------|--------------|
| Claude Opus 4.5 | 33.2%       | 32.2%             | 0.9%               | 88.2%        |
| Gemini 3 Pro    | 31.1%       | 30.3%             | 0.7%               | 90.1%        |
| GLM 4.7         | 28.4%       | 27.3%             | 0.9%               | 83.4%        |
| Gemini 3 Flash  | 26.1%       | 25.4%             | 0.7%               | 86.5%        |
| DeepSeek R1     | 26.6%       | 25.5%             | 1.1%               | 82.6%        |
| Kimi K2         | 23.5%       | 22.9%             | 0.7%               | 80.8%        |
| GPT 5.2         | 22.9%       | 22.4%             | 0.5%               | 85.4%        |
| QwQ 32B         | 25.1%       | 24.1%             | 0.9%               | 75.2%        |
| Qwen 2.5 32B    | 20.2%       | 19.3%             | 0.9%               | 72.7%        |
| Llama 3.1 8B    | 17.1%       | 16.3%             | 0.7%               | 66.5%        |

Table 23: Edge-level error patterns. Graph Match: percentage of generated edges matching ground-truth (STRONG + WEAK). Direction Correct/Reversed: among matched edges. Text Support: percentage with textual grounding (EXPLICIT + IMPLIED + GRAPH\_ONLY).

## P Node Count Specification

In ReCITE, models are provided with the expected number of nodes for each sample. This constraint is necessary because causal graphs allow for many valid levels of abstraction for the same underlying system. For example, A text describing economic relationships could yield a correct 5-node high-level graph or an equally correct 50-node version. Without specifying granularity, evaluation becomes ill-defined: we cannot distinguish models that correctly identify relationships at a different abstraction level from models that fail to identify relationships entirely. Real-world causal modeling tasks typically specify desired level of granularity, making this a realistic constraint rather than an artificial simplification. Additionally, in practice, LLM inference is inexpensive, so a practitioner can easily generate graphs across a range of node counts (e.g., 10, 20, 30, ...) and select the granularity that best fits their needs.

## Q 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  $\times$  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,  $F_1$  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.

## R Explicitness Labeling Prompt

Explicitness 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 *RI*

1544 (et al., 2025) to label the level of explicitness for  
 1545 each node in each sample when provide the ground  
 1546 truth and source text using the prompt below. To  
 1547 allow for fine-grained analysis of level of explicit-  
 1548 ness, we allow for three different levels of explic-  
 1549 itness for each node. (i) Explicit (it or a synonym  
 1550 of the node’s name appears in the text), (ii) Im-  
 1551 plicit (the node implicitly or indirectly appears in  
 1552 the text), or (iii) Absent (the node does not appear  
 1553 in the text whatsoever). We iteratively reject and  
 1554 retry any answers that do not meet formatting re-  
 1555 quirements until we receive valid answers for every  
 1556 benchmark sample to ensure that all samples have  
 1557 levels of explicitness. We detail the prompt used  
 1558 for this below. These node-level labels are used as  
 1559 the basis for calculation of level of explicitness.

**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 through 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": {
```

1560

```

    "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.

1561

S Variable Correction

1562

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

1563

1564

1565

1566

1567

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

1568

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` 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"  
  }  
]
```

```
}
```

T PDF-to-Markdown Conversion

We utilize an 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, 2025a)* 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 pre-processing 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

1570

1571

1572

1573

1574

1575

1576

1577

1578

1579

1580

1581

1582

1583

1584

1585

1586

1587

1588

1589

1590

1591

1592

1569

1593

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.

U Text Normalization Prompt

In order to ensure the realism of the ReCITE 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"},
  ...
]
}
```

1625

1626

V 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.

1627

W Standard Formatting Correction Prompt

1628

1629

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 its 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.

1630

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.

1631

X 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
```
```

1635

Y Name-Assisted Formatting Correction Prompt

1636

1637

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 its 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

1638

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
```
```

Z LLM-as-a-Judge Prompt

LLM-as-a-Judge Prompt

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 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, err 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.

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

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

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

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

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.

```
\verb|```yaml|
```

If evaluating node precision:

```
node_precision_evaluations:  
- node_number: <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>
```

```
...
```

1649