Causality Meets the Table: Debiasing LLMs for Faithful TableQA via Front-Door Intervention

Zhen Yang[1,2,3], Ziwei Du[1,2,3], Minghan Zhang[1,2,3], Wei Du[1,2,3], Jie Chen[1,2,3], Fulan Qian[1,2,3], Shu Zhao[1,2,3]*

[1]School of Computer Science and Technology, Anhui University
[2]Information Materials and Intelligent Sensing Laboratory of Anhui Province, Anhui University
[3]Anhui Provincial Key Laboratory of Security Artificial Intelligence, Anhui University
*zhaoshuzs2002@hotmail.com

Abstract

Table Question Answering (TableQA) combines natural language understanding and structured data reasoning, posing challenges in semantic interpretation and logical inference. Recent advances in Large Language Models (LLMs) have improved TableQA performance through Direct Prompting and Agent paradigms. However, these models often rely on spurious correlations, as they tend to overfit to token co-occurrence patterns in pretraining corpora, rather than perform genuine reasoning. To address this issue, we propose **C**ausal **I**ntervention **T**ableQA (CIT), which is based on a structural causal graph and applies front-door adjustment to eliminate bias caused by token co-occurrence. CIT formalizes TableQA as a causal graph and identifies token co-occurrence patterns as confounders. By applying front-door adjustment, CIT guides question variant generation and reasoning to reduce confounding effects. Experiments on multiple benchmarks show that CIT achieves state-of-the-art performance, demonstrating its effectiveness in mitigating bias. Consistent gains across various LLMs further confirm its generalizability. We release our code here.

1 Introduction

Tabular data is a prevalent type of structured information, commonly found in many fields [Yang et al., 2025, Lee et al., 2024, Xia et al., 2023]. Table Question Answering (TableQA), which aims to answer natural language questions over tables, plays a key role in decision support and data analysis. Early methods focused on SQL-based semantic parsing [Zhong et al., 2017] or pretraining on table-specific corpora [Ou and Liu, 2022, Eisenschlos et al., 2020, Xie et al., 2022]. More recently, Large Language Models (LLMs) have achieved strong results on TableQA by leveraging In-Context Learning [Sui et al., 2023, Chen, 2023a] and Chain-of-Thought (CoT) prompting [Cheng et al., 2023, Ye et al., 2023b]. Based on CoT, two main paradigms have emerged: Direct Prompting (DP), which performs natural language reasoning, and Agent, which relies on symbolic code execution.

Despite strong empirical performance, prompting strategies in LLM-based TableQA are often not robust [Ye et al., 2023a]. LLMs tend to rely on token co-occurrence patterns from pretraining data, which can lead to spurious correlations and unfaithful reasoning [Lyu et al., 2023, Wang et al., 2023c, Bao et al., 2024, Turpin et al., 2023]. For example, as shown in Figure 1, phrases like exactly frequently co-occur with answers like yes, causing LLMs to prefer yes even when no is correct.

*Corresponding authors

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

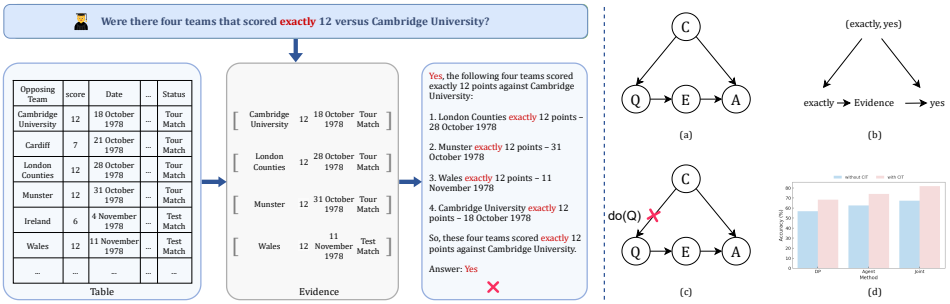

Figure 1: Illustration of token co-occurrence bias as a confounder in LLM-based TableQA. (a) Causal graph showing confounding in reasoning; (b) Real-world example where pretraining bias causes unfaithful answers; (c) Do-intervention to block the back-door path; (d) Adversarial example demonstrating spurious correlations.

This creates a confounder that affects both question interpretation and answer prediction, forming a back-door path that distorts reasoning. Although causal methods have been proposed to address such issues [Niu et al., 2021, Tian et al., 2022, Guo et al., 2023], they often rely on observable confounders or model internals, limiting their applicability in TableQA where confounders are typically latent.

Beyond the above qualitative analysis, we also conduct quantitative verification of these limitations. Specifically, we apply double negation perturbations to logically equivalent examples from TabFact dataset [Chen et al., 2020] and observe a substantial drop in accuracy. This suggests that LLMs rely more on surface-level linguistic patterns than on deep reasoning.

To address these challenges, we reinterpret TableQA from a causal perspective. As shown in Figure 1(a), ideal reasoning follows $Q \rightarrow E \rightarrow A$, but token co-occurrence introduces a confounder C, forming a spurious path $Q \leftarrow C \rightarrow A$. We adopt front-door adjustment [Pearl et al., 2016], which enables causal estimation using only observed variables, without requiring access to C or model internals. We propose Causal Intervention TableQA (CIT), a framework that mitigates bias from token co-occurrence patterns through front-door adjustment. CIT avoids explicit do-calculus by decomposing the adjustment process into four components: (1) *Question Variant Generation*, which produces diverse paraphrases of Q to reduce lexical bias; (2) *Evidence Aggregation*, which combines retrieved content across variants to improve coverage; (3) *Answer Inference*, which applies both DP and Agent reasoning strategies; (4) *Joint Voting*, which selects the final answer via majority voting. Experiments across multiple TableQA benchmarks and LLMs show that CIT consistently improves reasoning robustness and generalization. Our main contributions are as follows:

- **Causal formulation of co-occurrence bias**: We are the first to introduce causal intervention into LLM-based TableQA by modeling token co-occurrence as latent confounding and applying front-door adjustment to mitigate its effect.
- **Efficient intervention via question variants:** We estimate causal effects using semantically diverse question variants. A single-pass generation strategy ensures low overhead.
- **Broad empirical validation:** CIT achieves state-of-the-art performance on multiple datasets across both open- and closed-source LLMs, demonstrating strong generalization.

2 Related Work

2.1 LLM-based TableQA

Recent advances in large language models (LLMs) have greatly improved TableQA performance by leveraging general reasoning capabilities [Pal et al., 2023, Lee et al., 2024, Zhong et al., 2017, Yang et al., 2025]. Existing methods mainly follow two paradigms: Direct Prompting, which guides reasoning in a single step [Sui et al., 2023, Chen, 2023a], and Agent, which decomposes the task into symbolic operations [Li et al., 2024b, Lei et al., 2023]. Representative methods are listed in Appendix A. However, these approaches focus primarily on guiding LLM reasoning, while

overlooking a key issue: LLMs often encode token co-occurrence patterns from pretraining data, which can induce spurious correlations between the question and the answer.

2.2 Causal Intervention

Causal inference offers a principled framework for addressing bias through interventions [Pearl et al., 2016, Pearl, 2019, Ren et al., 2023a,b]. Prior work has applied counterfactual [Niu et al., 2021, Xu et al., 2023, Yang et al., 2023b], back-door adjustment [Tian et al., 2022, Zhu et al., 2023], and front-door adjustment [Yang et al., 2021, Zhang et al., 2024a, Yang et al., 2023a] to mitigate spurious correlations. Recent studies have extended ideas to LLMs [Jin et al., 2023, Lyu et al., 2024], although many rely on heuristics or simplified causal graphs [Wang et al., 2023b, Tang et al., 2023]. In contrast to back-door methods that require explicit modeling of confounders, which is often infeasible for LLMs, front-door adjustment enables causal estimation using only observed variables. This makes it particularly suitable for LLM-based TableQA.

3 Preliminaries

3.1 TableQA

Given a question Q and a table T, an LLM first interprets the question, retrieves relevant evidence $E \subseteq T$ based on Q, and then reasons over Q and E to produce the final answer A. This process can be formally expressed as Equation 1:

$$E = Prompt_{retrieve}(Q, T), \quad A = Prompt_{answer}(Q, E) \tag{1}$$

Here, $Prompt_{retrieve}$ refers to the step where the LLM selects evidence based on the question. $Prompt_{answer}$ performs reasoning over the question and the evidence to get the answer.

3.2 Structural Causal Model (SCM)

Causal inference offers a framework for modeling interventions and estimating causal effects. A central tool is the Structural Causal Model (SCM)[Pearl et al., 2016], which represents dependencies among variables as a directed acyclic graph (DAG) $G = (V, E)$, where V is the set of nodes and E is the set of edges. In Figure 1, we illustrate the causal graph constructed for the TableQA task and explain its components as follows.

$Q \rightarrow E \rightarrow A$. In TableQA, the input question Q determines the selection of evidence E, which in turn leads to the answer A. This forms the ideal causal path $Q \rightarrow E \rightarrow A$.

$Q \leftarrow C \rightarrow A$. During pretraining, LLMs tend to overfit to token co-occurrence patterns in the corpus. In certain cases, this behavior interferes with reasoning and leads to biased predictions. We treat this as a latent confounder, denoted as C, which introduces a spurious back-door path $Q \leftarrow C \rightarrow A$.

$do(Q)$. To identify the true causal effect, it is necessary to block the influence of the confounder C using the do-operator [Fenton et al., 2020]. In an SCM, applying $do(Q)$ corresponds to removing all incoming edges to Q, thereby eliminating the indirect effect of C on A through Q. In causal inference, the do-operator represents an ideal intervention that forcibly sets the value of a variable. However, such interventions are typically infeasible in observational data. Therefore, techniques such as front-door or back-door adjustment are commonly used to estimate causal effects without directly applying the do-operator.

3.3 Front-door Adjustment

Back-door adjustment requires access to the confounder C, which is unobservable in LLMs. In contrast, front-door adjustment bypasses this need and is thus more applicable. By the law of total probability, the interventional distribution is given in Equation 2.

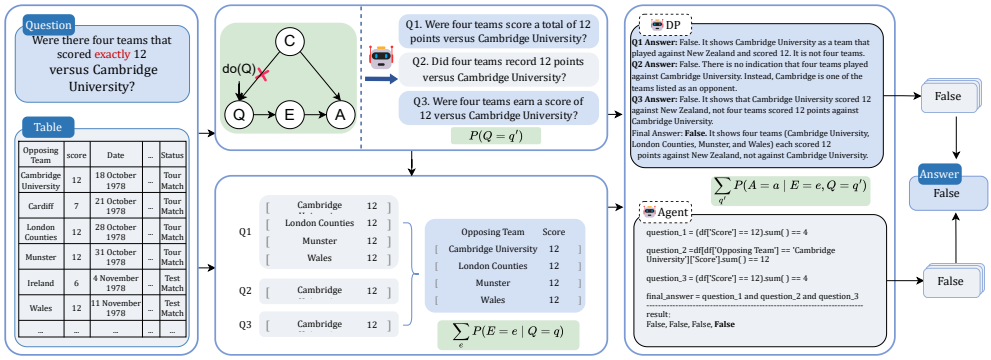

Figure 2: Overview of the CIT. Given a question and table, CIT derives the final answer through: (1) Question Variant Generation, which produces semantically diverse variants; (2) Evidence Aggregation, which extracts and unifies evidence across variants; (3) Answer Inference, which combines Direct Prompting and Agent; and (4) Joint Voting, which selects the final answer via majority voting.

$$P(A \mid do(Q)) = \sum_e P(A = a \mid do(Q = q), E = e)\, P(E = e \mid do(Q = q)) \tag{2}$$

By applying the law of total probability and the assumptions of SCM, this expression can be further transformed into the final formula shown in Equation 3. The full derivation is provided in Appendix B.

$$P(A = a \mid do(Q = q)) = \sum_e P(E = e \mid Q = q) \sum_{q'} P(A = a \mid E = e, Q = q')\, P(Q = q') \tag{3}$$

4 Method

By applying the do-operator, we block the bias introduced by the confounder C. Through front-door adjustment, this intervention can be reformulated into a tractable expression using only observed variables. Based on Equation 3, our method consists of four components: (1) Question Variant Generation, (2) Evidence Aggregation, (3) Answer Inference, and (4) Joint Voting. We describe each component in detail below.

4.1 $P(Q = q')$: Question Variant Generation

To estimate $P(Q = q')$, we generate a set of semantically equivalent question variants $\{q'_i\}_{i=1}^n$ that preserve the intent of the original question Q while differing in surface form. To reduce the cost of LLM inference, we adopt a single-pass generation strategy using a prompt-based generator, denoted as $Prompt_{gen}$, which produces all variants in a single call, as shown in Equation 4. Here, Q is the input question, T is the table, and the prompt used for variant generation is detailed in Appendix C.

$$\{q'_i\}_{i=1}^n = LLM(Q, T, Prompt_{gen}) \tag{4}$$

Since all variants are generated simultaneously, their generation probability is approximated as uniform. As a result, $P(Q = q')$ is treated as a constant and omitted from the final formulation.

Beyond the causal perspective, our method can also be understood semantically. When encoded by an LLM, the original question and its variants are represented as high-dimensional vectors in a shared semantic space. Generating multiple variants effectively samples points around the original question, forming a dense semantic neighborhood. This encourages the LLM to reason over meaning rather than surface form, which improves robustness and helps mitigate bias.

4.2 $\sum_e P(E = e \mid Q = q)$: Evidence Aggregation

To estimate $\sum_e P(E = e \mid Q = q)$, we extract supporting evidence for the given question. Since CIT considers multiple question variants $\{q'_i\}$, evidence must be extracted for each variant. To reduce the cost, we adopt a separate single-pass strategy using $Prompt_{retrieve}$ for evidence extraction. All question variants are provided to the LLM in one prompt, and the LLM extracts the corresponding evidence $e_{q'_i}$ for each variant and then aggregate to form the final evidence, as shown in Equation 5. The prompt used by $Prompt_{retrieve}$ is detailed in Appendix C.

$$LLM(T, q, q', Prompt_{retrieve}) \rightarrow e_{q'_i}, \quad e = \bigcup_{i=1}^{n} e_{q'_i} \tag{5}$$

This union operation ensures that all potentially useful evidence across diverse paraphrases of the question is captured. By aggregating information from multiple linguistic perspectives, we construct a more complete and robust foundation for downstream reasoning.

4.3 $\sum_{q'} P(A = a \mid E = e, Q = q')P(Q = q')$: Answer Inference

To estimate $\sum_{q'} P(A = a \mid E = e, Q = q')P(Q = q')$, we infer under each variant q'. We consider two reasoning paradigms commonly used in LLM-based TableQA: Direct Prompting and Agent.

Direct Prompting (DP) Reasoning. DP guides the LLM to generate step-by-step reasoning via CoT to obtain the answer, improving performance on complex TableQA tasks. For each question variant q', the LLM generates a token sequence conditioned on q' and the retrieved evidence e, modeled autoregressively as Equation 6.

$$P(A = a^{DP} \mid E = e, Q = q') = \prod_{l=1}^{L} P(w_l \mid w_{<l}, e, q') \tag{6}$$

Here, w_l is the l-th token, $w_{<l}$ the preceding tokens, and L the sequence length. The LLM maximizes this joint probability to generate reasoning steps and the final answer a^{DP}. However, this formulation may overfit to frequent patterns from pretraining, leading to overconfident but unfaithful predictions. Our causal framework mitigates this by aggregating outputs across diverse question variants.

Symbolic Reasoning with Agent. In contrast to natural language reasoning, Agent allows the LLM to generate executable Python code for structured operations over table evidence. Given a question variant q' and evidence e, the LLM produces a code snippet $code(q', e)$, and the final answer is obtained by executing it within a Python shell as Equation 7. Here $execute$ denotes run the code.

$$P(A = a^{Agent} \mid E = e, Q = q') = execute(code(q', e)) \tag{7}$$

This symbolic approach allows direct operations on tabular as filtering, aggregation, and arithmetic—enabling precise numerical. To reduce API cost, we adopt a unified execution strategy: the original question q and its variants set$\{q'_i\}$ are processed in a single LLM call. The model reasons over each variant independently and aggregates intermediate results into a final answer a^{Agent}. This one-shot process improves efficiency and ensures semantic consistency across variants.

4.4 Joint Voting

CIT supports both DP and Agent reasoning, and integrates their outputs via majority voting to exploit their complementary strengths. For each question q, the framework performs n rounds of DP and m rounds of Agent reasoning, yielding answer sets $\mathcal{A}^{\text{DP}} = \{a_1^{\text{DP}}, \ldots, a_n^{\text{DP}}\}$ and $\mathcal{A}^{\text{Agent}} = \{a_1^{\text{Agent}}, \ldots, a_m^{\text{Agent}}\}$. The final prediction is selected by majority vote over both sets as Equation 8:

$$P(A = a \mid E = e, Q = q') = Majority\ Vote(A^{DP} \cup A^{Agent}) \tag{8}$$

In case of a tie, one of the top answers is selected uniformly at random. The hyperparameters n and m control the number of DP and Agent rounds. We vary (n, m) in ablations to explore performance-cost

trade-offs. Beyond voting, this joint design enhances semantic diversity by reasoning over multiple variants q' and evidence sets E. This is especially beneficial for ambiguous or biased questions, where aggregating diverse reasoning paths helps mitigate confounder-induced errors.

5 Experimental Setup

5.1 Datasets and Evaluation

Dataset. We evaluate on three datasets: WikiTableQuestions (WTQ) [Pasupat and Liang, 2015], TabFact[Chen et al., 2020], and FetaQA [Nan et al., 2022]. WTQ involves aggregation, comparison, and arithmetic reasoning, with 4,344 test examples. TabFact is a fact verification task over 2,024 samples. FetaQA features free-form questions that require integrating information on 2,003 samples.

Evaluation. Following prior work [Liu et al., 2024a, Yang et al., 2025], we use exact match accuracy for WTQ and TabFact, which focus on short-form answers. For FetaQA, which requires long-form generation, we report BLEU [Papineni et al., 2002] to evaluate answer quality.

5.2 Implementation Details

To ensure fair comparison, we first evaluate CIT using GPT-3.5 as LLM. To assess generalizability, we further test CIT across LLMs: *Open-source*: LLaMA 2-7B/13B/70B, DeepSeek-R1; *Closed-source*: GPT-3.5, GPT-4, GLM 4, Gemini 1.5, Claude 3.5. All LLMs temperature is 0.8.

Table 1: Results on WikiTableQuestions with GPT-3.5. CIT-DP and CIT-Agent show results for DP and Agent modes separately. CIT-DP&Agent shows the joint voting result.

Method	Acc.
OmniTab (22' NAACL)	61.30
Codex SQL (23' ICLR)	61.10
BINDER (23' ICLR)	64.60
DATER (23' SIGIR)	65.90
DTE (23' ACL)	54.20
TACR (23' arXiv)	60.20
ITR (23' ACL)	63.40
StructGPT (23' EMNLP)	57.00
Liu et al. (24' arXiv)	55.80
Cabinet (24' ICLR)	69.10
CHAIN-OF-TABLE (24' ICLR)	59.94
ReAcTable (24' VLDB)	68.00
SYNTQA (24' EMNLP)	70.40
Mix-SC (DP&Agent) (24' NAACL)	73.65
TIDE (DP&Agent) (25' ICLR)	75.00
CIT-DP	65.40
CIT-Agent	73.76
CIT-DP&Agent	**76.38**

Table 2: Results on TabFact dataset with GPT-3.5. CIT-DP and CIT-Agent show results for DP and Agent modes separately. CIT-DP&Agent shows the joint voting result.

Method	Acc.
TAPAS-large (20' EMNLP)	81.00
TAPEX-large (21' ICLR)	84.20
SaMOE (22' ACL)	86.70
SASP (22' ACL)	77.00
T5-3B (22' EMNLP)	83.68
Codex end-to-end (23' ICLR)	72.60
Codex SQL (23' ICLR)	80.70
BINDER (23' ICLR)	85.10
DATER (23' SIGIR)	85.60
StructGPT (23' EMNLP)	87.30
CHAIN-OF-TABLE (24' ICLR)	80.20
ReAcTable (24' VLDB)	86.10
Tab-PoT (24' arXiv)	85.77
Mix-SC (DP&Agent) (24' NAACL)	88.50
TIDE (DP&Agent) (25' ICLR)	89.82
CIT-DP	83.15
CIT-Agent	90.61
CIT-DP&Agent	**91.30**

5.3 Baselines

We compare CIT with pretraining models and LLM-based methods, include SASP [Ou and Liu, 2022], TAPAS-large [Eisenschlos et al., 2020], T5-3B [Xie et al., 2022], TAPEX-large [Liu et al., 2021], Task Configs [Chen et al., 2023], TARGET [Ji et al., 2024], TabCot [Chen, 2023b], TAG-QA [Zhao et al., 2023a], UniTabPT[Sarkar and Lausen, 2023], and Codex [Cheng et al., 2023], BINDER [Cheng et al., 2023], DATER [Ye et al., 2023b], StructGPT [Jiang et al., 2023], DTE [Wang et al., 2023a], TACR [Wu et al., 2023], ITR [Lin et al., 2023], Tab-PoT [Xiao et al., 2024], [Liu et al., 2024a],

CHAIN-OF-TABLE [Wang et al., 2024], ReAcTable [Zhang et al., 2024c], Cabinet [Patnaik et al., 2024], SYNTQA [Zhang et al., 2024b], [Liu et al., 2024b] and TIDE [Yang et al., 2025]. Details of the baseline implementations are provided in Appendix D.

6 Results and Analysis

6.1 Main Results

Table 1 shows that CIT achieves state-of-the-art performance on WikiTableQuestions, improving the previous best by 2.73%. On TabFact (Table 2), it outperforms the strongest baseline by 2.21%, a relative gain of 21.71% over the error rate. CIT also leads clearly on FetaQA (Table 3). We analyze the effectiveness of CIT from both qualitative and quantitative perspectives: *Qualitative Perspective:* CIT introduces causal reasoning via a structural causal model, enabling identification and blocking of confounding bias through front-door adjustment. Question variants help mitigate spurious correlations from pretraining, improving answer faithfulness. *Quantitative Perspective:* On adversarial data with double negation (Figure 1), CIT significantly outperforms non-intervention models, confirming its robustness. The combination of DP and Agent leverages complementary strengths and reduces reliance on any single reasoning mode.

Table 3: Results on FetaQA with GPT-3.5.

Methods	**BLEU(%)**
Task Configs (23' ACL)	27.80
T5-large (23' SIGIR)	30.54
TAG-QA (23' ACL)	31.84
UniTabPT (23' NeurIPS)	33.12
Codex (23' SIGIR)	27.96
DATER (23' SIGIR)	30.92
TabCot (23' EACL)	29.36
TARGET (24' NeurIPS)	24.13
ReAcTable (24' VLDB)	30.43
CIT-DP	36.15
CIT-Agent	33.65
CIT-DP&Agent	**36.34**

Table 4: Impact of answer selection in CIT-DP and CIT-Agent.

Agent	DP	WTQ	TabFact	FetaQA
1	1	61.60	88.34	35.68
3	3	66.11	90.42	35.93
5	5	66.92	90.81	37.15
1	3	64.46	85.42	35.98
3	1	61.14	89.18	35.73
1	5	69.38	82.16	36.17
5	1	71.04	87.35	36.03
3	5	66.62	86.51	35.99
5	3	**76.38**	**91.30**	**36.34**

We also observe that CIT-Agent consistently outperforms CIT-DP on WTQ and TabFact, likely due to its structured execution over tables and ability to handle large inputs with precise symbolic operations. On FetaQA, however, CIT-DP performs comparably, as BLEU evaluation favors the fluency of natural language outputs generated by DP, which better align with reference answers.

6.2 Effect of n and m in Answer Aggregation

To reduce bias from single-pass or single-mode inference, CIT performs n rounds of Direct Prompting and m rounds of Agent reasoning, aggregating all $n + m$ results via majority voting. As shown in Table 4, performance improves with larger n or m. More Agent results tend to yield better accuracy, while DP offers complementary signals. These results underscore the value of reasoning diversity and validate the design of our causal aggregation framework.

6.3 Ablation Study

To assess the contribution of each component, we conduct ablation studies under both DP and Agent. We consider two variants: (1) *w/o Evidence Aggregation*, which uses evidence extracted from the original question q for all variants; and (2) *w/o Question Variants*, which disables variant generation and reasons directly over q. Results in Table 5 show that removing evidence aggregation yields a minor drop, as most variants retrieve similar evidence. In contrast, removing question variants leads to a significant decline, confirming their importance in mitigating confounding bias. These findings validate the role of front-door variant generation in enabling robust, causally grounded inference.

Table 5: Ablation results on datasets.

Method	WTQ	TabFact	FetaQA
CIT-DP	**66.40**	**83.15**	**33.15**
w/o Question Variants	62.66 (↓ 3.74)	76.38 (↓ 6.77)	31.71 (↓ 1.44)
w/o Evidence Aggregation	63.31 (↓ 3.09)	79.64 (↓ 3.51)	32.43 (↓ 0.72)
CIT-Agent	**73.76**	**90.61**	**36.15**
w/o Question Variants	68.39 (↓ 5.37)	86.86 (↓ 3.75)	34.21 (↓ 1.94)
w/o Evidence Aggregation	71.39 (↓ 2.37)	87.99 (↓ 2.62)	35.38 (↓ 0.77)
CIT-DP&Agent	**76.38**	**91.30**	**36.34**
w/o Question Variants	71.50 (↓ 4.88)	89.87 (↓ 1.43)	35.41 (↓ 0.93)
w/o Evidence Aggregation	74.47 (↓ 1.91)	90.27 (↓ 1.03)	36.02 (↓ 0.32)

6.4 Generalization Across LLMs

We evaluate CIT on a diverse set of open-source and closed-source LLMs, using the same setup as Section 5.2. As shown in Table 6, CIT consistently improves performance across all models, regardless of size, architecture, or pretraining corpus. These results highlight the transferability of our causal intervention framework and suggest that core TableQA challenges—semantic ambiguity, spurious correlations, and evidence selection—are shared across LLMs.

Table 6: Comparison of LLMs with and without CIT.

	Models	**Init Accuracy(%)**	**+ CIT Accuracy(%)**
Open-source	Llama 2-7b	48.34	49.95 (**1.61** ↑)
	Llama 2-13b	50.18	52.07 (**1.89** ↑)
	Llama 2-70b	59.02	61.33 (**2.31** ↑)
	DeepSeek R1	78.38	80.48 (**2.10** ↑)
Closed-source	GLM 4	65.84	66.53 (**0.69** ↑)
	GPT 4	70.89	77.09 (**6.20** ↑)
	Gemini 1.5	61.56	66.51 (**4.95** ↑)
	Claude 3.5	72.33	75.87 (**3.54** ↑)

6.5 Analysis of Influential Factors

Effect of Variant Quantity. As shown in Figure 3, accuracy improves with more question variants due to greater semantic coverage, but saturates beyond three. Token usage also increases sharply, especially during reasoning. We set the default to three variants to balance performance and efficiency.

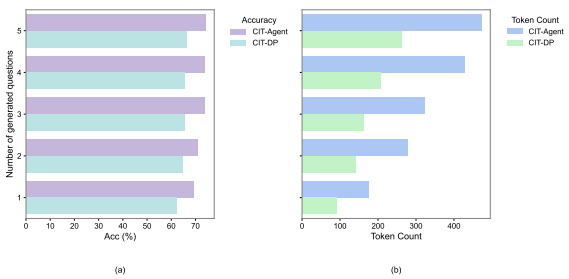

Figure 3: Changes in accuracy and token consumption under different numbers of question variants.

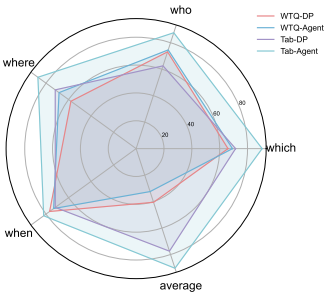

Figure 4: Impact of question type on CIT performance.

Performance Across Question Types We analyze CIT across question types on WTQ and TabFact (Figure 4), excluding FetaQA due to BLEU's incompatibility with discrete categories. Questions are grouped by keywords (e.g., *who*, *when*, *average*). On TabFact, both modes perform well across types. On WTQ, performance drops on numerical reasoning, especially aggregation. CIT-Agent further struggles due to hallucinated code constraints (e.g., unnecessary `unique()`). Despite this, the two modes show complementary strengths, supporting our joint reasoning.

Impact of LLM Size. We evaluate CIT using LLaMA 2 models of different sizes. As shown in Table 6, larger LLMs consistently yield better performance, likely due to their enhanced knowledge representation and reasoning capabilities. These strengths improve both the quality of question variants and the accuracy of evidence selection, which are critical to the effectiveness of front-door adjustment. This suggests that CIT benefits from scaling and can serve as a lightweight debiasing layer for increasingly powerful LLMs.

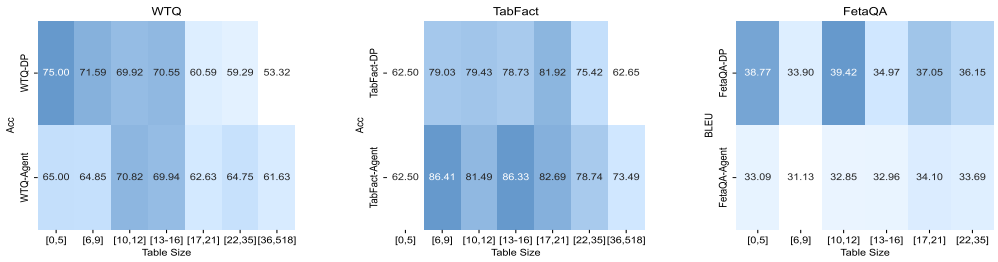

Figure 5: Impact of table size on TableQA performance.

Table Size Sensitivity. We evaluate CIT's robustness under varying table sizes by grouping test samples into bins of roughly 430 examples and computing average accuracy per bin (Figure 5). While accuracy generally decreases with larger tables, CIT remains stable overall. CIT-Agent outperforms CIT-DP on larger tables, as it executes Python code over full tables, bypassing context length limits, whereas DP's reliance on in-context reasoning leads to degraded performance with long inputs.

Efficiency and API Usage. CIT is efficient and compatible with both open-source and closed-source LLMs. As shown in Table 6, it performs well on non-API models like LLaMA and DeepSeek, supporting private deployment. For API-based use, CIT requires only three calls: one each for variant generation, evidence integration, and answer inference. Table 7 shows that CIT achieves strong performance with substantially fewer API calls than prior LLM-based methods.

Table 7: Comparison of methods with results and API calls

Methods	Result	Number of API calls
CHAIN-OF-TABLE	59.94	(Next Operation 1 + Argument 1 + Transform 1) * Iter $N = 3N$
CIT	76.38	Generate Questions 1 + Evidence Integration 1 + Answer 1 = 3

Error Case Analysis. We manually examine 100 examples to identify common sources of failure. For CIT-DP, most errors arise from incorrect answer formatting and the inability to recognize special table lines such as headers, footnotes, or merged cells. For CIT-Agent, errors often involve hallucinated constraints—such as adding non-existent conditions—and occasional format inconsistencies. These findings highlight the challenges of aligning LLM output with task-specific answer expectations. A detailed breakdown of error types and representative examples is provided in Appendix F.

Additional Analyses. We further investigate whether CIT is affected by LLM data contamination, details are reported in Appendix E. We also enumerate the range of question types that CIT can handle effectively—including *where*, *when*, *which*, *what*, *who*, *is/does*, *how many*, *average*, *sum*, etc.—with corresponding case examples shown in Appendix G.

7 Limitations

While CIT offers a principled way to mitigate confounding bias, its effectiveness depends on the quality of question variants. Limited diversity or semantic inconsistency may hinder coverage of the original intent. Future work may explore controlled generation or filtering to improve variant quality.

8 Conclusion

We present CIT, a causal intervention framework for TableQA that applies front-door adjustment to mitigate latent confounding bias in LLM-based reasoning. By modeling TableQA as a structural causal process, CIT identifies and blocks spurious back-door paths introduced by pretraining. The method implements this via question variant generation, evidence aggregation, and joint reasoning with both Direct Prompting and Agent paradigms. Extensive results across multiple benchmarks and LLMs demonstrate CIT's robustness, effectiveness, and generality.

9 Acknowledgements

Our work is supported by the National Natural Science Foundation of China (62476003), Anhui Province Excellent Scientific Research and Innovation Team (2024AH010004), Anhui Provincial Natural Science Foundation - Water Science Joint Fund (2408055US006), the University Synergy Innovation Program of Anhui Province (GXXT-2023-050), and SMP-Zhipu.AI Large Model Cross-Disciplinary Fund (SMP-Zhipu20240210). We also acknowledge the support from Zhipu AI-Anhui University Joint Research Center, and the High-Performance Computing Platform of Anhui University.

References

Guangsheng Bao, Hongbo Zhang, Linyi Yang, Cunxiang Wang, and Yue Zhang. Llms with chain-of-thought are non-causal reasoners. *arXiv e-prints*, pages arXiv–2402, 2024.

Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Jaydeep Sen, Mustafa Canim, Soumen Chakrabarti, Alfio Gliozzo, and Karthik Sankaranarayanan. Topic transferable table question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4159–4172, 2021.

Jifan Chen, Yuhao Zhang, Lan Liu, Rui Dong, Xinchi Chen, Patrick Ng, William Yang Wang, and Zhiheng Huang. Improving cross-task generalization of unified table-to-text models with compositional task configurations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5523–5539, 2023.

Wenhu Chen. Large language models are few (1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, 2023a.

Wenhu Chen. Large language models are few (1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, 2023b.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *The 8th International Conference on Learning Representations*, 2020.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. Binding language models in symbolic languages. In *The 11th International Conference on Learning Representations*, 2023.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. Tables as images? exploring the strengths and limitations of llms on multimodal representations of tabular data. *arXiv preprint arXiv:2402.12424*, 2024.

Julian Eisenschlos, Syrine Krichene, and Thomas Mueller. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, 2020.

Norman E. Fenton, Martin Neil, and Anthony C. Constantinou. The book of why: The new science of cause and effect, judea pearl, dana mackenzie. basic books (2018). *Artif. Intell.*, 284:103286, 2020.

Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1978–1988, 2020.

Wangzhen Guo, Qinkang Gong, Yanghui Rao, and Hanjiang Lai. Counterfactual multihop qa: A cause-effect approach for reducing disconnected reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4214–4226, 2023.

Xingyu Ji, Aditya Parameswaran, and Madelon Hulsebos. Target: Benchmarking table retrieval for generative tasks. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, 2023.

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

Kezhi Kong, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Chuan Lei, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Opentab: Advancing large language models as open-domain table reasoners. In *The 12th International Conference on Learning Representations*, 2024.

Younghun Lee, Sungchul Kim, Ryan A Rossi, Tong Yu, and Xiang Chen. Learning to reduce: Towards improving performance of large language models on structured data. In *First Workshop on Long-Context Foundation Models@ ICML 2024*, 2024.

Fangyu Lei, Tongxu Luo, Pengqi Yang, Weihao Liu, Hanwen Liu, Jiahe Lei, Yiming Huang, Yifan Wei, Shizhu He, Jun Zhao, et al. Tableqakit: A comprehensive and practical toolkit for table-based question answering. *arXiv preprint arXiv:2310.15075*, 2023.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and ZHAO-XIANG ZHANG. Sheetcopilot: Bringing software productivity to the next level through large language models. *Advances in Neural Information Processing Systems*, 36, 2024a.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. Table-gpt: Table fine-tuned gpt for diverse table tasks. *Proceedings of the ACM on Management of Data*, 2(3):1–28, 2024b.

Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adrià de Gispert, and Gonzalo Iglesias. An inner table retriever for robust table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9909–9926, 2023.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. Tapex: Table pre-training via learning a neural sql executor. *arXiv preprint arXiv:2107.07653*, 2021.

Tianyang Liu, Fei Wang, and Muhao Chen. Rethinking tabular data understanding with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 450–482, 2024a.

Yujian Liu, Jiabao Ji, Tong Yu, Ryan Rossi, Sungchul Kim, Handong Zhao, Ritwik Sinha, Yang Zhang, and Shiyu Chang. Augment before you try: Knowledge-enhanced table question answering via table expansion. *arXiv preprint arXiv:2401.15555*, 2024b.

Tongxu Luo, Fangyu Lei, Jiahe Lei, Weihao Liu, Shihu He, Jun Zhao, and Kang Liu. Hrot: Hybrid prompt strategy and retrieval of thought for table-text hybrid question answering. *arXiv preprint arXiv:2309.12669*, 2023.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, 2023.

Zhiheng Lyu, Zhijing Jin, Fernando Gonzalez, Rada Mihalcea, Bernhard Schölkopf, and Mrinmaya Sachan. On the causal nature of sentiment analysis. *arXiv e-prints*, pages arXiv–2404, 2024.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, et al. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12700–12710, 2021.

Suixin Ou and Yongmei Liu. Learning to generate programs for table fact verification via structure-aware semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7624–7638, 2022.

Vaishali Pal, Andrew Yates, Evangelos Kanoulas, and Maarten de Rijke. Multitabqa: Generating tabular answers for multi-table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6322–6334, 2023.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318, 2002.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, 2015.

Sohan Patnaik, Heril Changwal, Milan Aggarwal, Sumit Bhatia, Yaman Kumar, and Balaji Krishnamurthy. Cabinet: Content relevance-based noise reduction for table question answering. In *The 12th International Conference on Learning Representations*, 2024.

Judea Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Lin Ren, Yongbin Liu, Yixin Cao, and Chunping Ouyang. Covariance-based causal debiasing for entity and relation extraction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 2627–2640. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.FINDINGS-EMNLP.173. URL https://doi.org/10.18653/v1/2023.findings-emnlp.173.

Lin Ren, Yongbin Liu, and Chunping Ouyang. Causal inference-based debiasing framework for knowledge graph completion. In Terry R. Payne, Valentina Presutti, Guilin Qi, María Poveda-Villalón, Giorgos Stoilos, Laura Hollink, Zoi Kaoudi, Gong Cheng, and Juanzi Li, editors, *The Semantic Web - ISWC 2023 - 22nd International Semantic Web Conference, Athens, Greece, November 6-10, 2023, Proceedings, Part I*, volume 14265 of *Lecture Notes in Computer Science*, pages 328–347. Springer, 2023b. doi: 10.1007/978-3-031-47240-4_18. URL https://doi.org/10.1007/978-3-031-47240-4_18.

Soumajyoti Sarkar and Leonard Lausen. Testing the limits of unified sequence to sequence llm pretraining on diverse table data tasks. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

Yuan Sui, Jiaru Zou, Mengyu Zhou, Xinyi He, Lun Du, Shi Han, and Dongmei Zhang. Tap4llm: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning. *arXiv preprint arXiv:2312.09039*, 2023.

Ziyi Tang, Ruilin Wang, Weixing Chen, Keze Wang, Yang Liu, Tianshui Chen, and Liang Lin. Towards causalgpt: A multi-agent approach for faithful knowledge reasoning via promoting causal consistency in llms. *arXiv preprint arXiv:2308.11914*, 2023.

Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. Debiasing nlu models via causal intervention and counterfactual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11376–11384, 2022.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

Bing Wang, Yan Gao, Zhoujun Li, and Jian-Guang Lou. Know what i don't know: Handling ambiguous and unknown questions for text-to-sql. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5701–5714, 2023a.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. A causal view of entity bias in (large) language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15173–15184, 2023b.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*, 2023c.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The 12th International Conference on Learning Representations*, 2024.

Jian Wu, Yicheng Xu, Yan Gao, Jian-Guang Lou, Börje F Karlsson, and Manabu Okumura. Tacr: A table-alignment-based cell-selection and reasoning model for hybrid question-answering. *arXiv preprint arXiv:2305.14682*, 2023.

Renqiu Xia, Bo Zhang, Haoyang Peng, Ning Liao, Peng Ye, Botian Shi, Junchi Yan, and Yu Qiao. Structchart: Perception, structuring, reasoning for visual chart understanding. *arXiv preprint arXiv:2309.11268*, 2023.

Bin Xiao, Burak Kantarci, Jiawen Kang, Dusit Niyato, and Mohsen Guizani. Efficient prompting for llm-based generative internet of things. *arXiv preprint arXiv:2406.10382*, 2024.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631, 2022.

Weizhi Xu, Qiang Liu, Shu Wu, and Liang Wang. Counterfactual debiasing for fact verification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6777–6789, 2023.

Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12996–13010, 2021.

Zhen Yang, Yongbin Liu, and Chunping Ouyang. Causal intervention-based few-shot named entity recognition. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15635–15646. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.FINDINGS-EMNLP.1046. URL https://doi.org/10.18653/v1/2023.findings-emnlp.1046.

Zhen Yang, Yongbin Liu, Chunping Ouyang, Lin Ren, and Wen Wen. Counterfactual can be strong in medical question and answering. *Inf. Process. Manag.*, 60(4):103408, 2023b. doi: 10.1016/J.IPM.2023.103408. URL https://doi.org/10.1016/j.ipm.2023.103408.

Zhen Yang, Ziwei Du, Minghan Zhang, Wei Du, Jie Chen, Zhen Duan, and Shu Zhao. Triples as the key: Structuring makes decomposition and verification easier in LLM-based tableQA. In *The Thirteenth International Conference on Learning Representations*, 2025.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023a.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 174–184, 2023b.

Congzhi Zhang, Linhai Zhang, and Deyu Zhou. Causal walk: Debiasing multi-hop fact verification with front-door adjustment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19533–19541, 2024a.

Siyue Zhang, Luu Anh Tuan, and Chen Zhao. Syntqa: Synergistic table-based question answering via mixture of text-to-sql and e2e tqa. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2352–2364, 2024b.

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M Patel. Reactable: Enhancing react for table question answering. *Proceedings of the VLDB Endowment*, 17(8):1981–1994, 2024c.

Wenting Zhao, Ye Liu, Yao Wan, Yibo Wang, Zhongfen Deng, and S Yu Philip. Localize, retrieve and fuse: A generalized framework for free-form question answering over tables. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 1–12, 2023a.

Yilun Zhao, Haowei Zhang, Shengyun Si, Linyong Nan, Xiangru Tang, and Arman Cohan. Investigating table-to-text generation capabilities of llms in real-world information seeking scenarios. *arXiv preprint arXiv:2305.14987*, 2023b.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

Jiazheng Zhu, Shaojuan Wu, Xiaowang Zhang, Yuexian Hou, and Zhiyong Feng. Causal intervention for mitigating name bias in machine reading comprehension. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12837–12852, 2023.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly stated in these two sections that the paper focuses on LLM-based TableQA, and we have outlined our contributions in the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We have discussed the limitations of our method in the Limitations section, including unresolved issues and potential future directions.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide theoretical explanations and formula derivations in the Preliminaries section.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We present the datasets, evaluation metrics, and experimental setup in the Experiments section, with additional prompt context details provided in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We conducted experiments using publicly available datasets with proper citations in the Experiments section. Additionally, we have included the critical prompt context from our code in the Appendix for reproducibility.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In the Experiments section, we detail the datasets, evaluation metrics, and experimental configurations, with complementary prompt context provided in the Appendix to ensure full reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In the Experiments section, we report averaged results across multiple runs, accompanied by comprehensive analysis and discussion in the main text.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: In the Experiments section, our primary evaluations were conducted through API calls. We provide detailed documentation of API call counts, token count and response times in the Results and Analysis section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The manuscript maintains full anonymity, with no author-identifying information disclosed in any section of the paper.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In the Experiments section, we conduct both positive and negative case analyses for our main experiments, with further examination provided in the Results and Analysis section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The study exclusively utilizes publicly available datasets and accesses publicly released large language models (LLMs) through their official APIs.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This work exclusively utilizes publicly available datasets, with proper citations provided in the Experiments section.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Detailed specifications of all LLMs used in this work are comprehensively documented in both the Experiments and Results and Analysis sections.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Related Work: LLM-based TableQA

Direct Prompting in TableQA. In Direct Prompting (DP), LLMs perform step-by-step reasoning through serialized natural language prompts, often under the Chain-of-Thought (CoT) framework [Kong et al., 2024, Zhao et al., 2023b, Deng et al., 2024]. Early DP methods [Zhao et al., 2023a, Sui et al., 2023, Chemmengath et al., 2021] used few-shot examples, SQL-style prompts, or zero-shot CoT to help LLMs decompose and solve complex queries. For instance, Luo et al.[Luo et al., 2023] constructed CoT exemplars with retrieval-based reconstruction, while BINDER[Cheng et al., 2023] composed sub-queries via SQL logic. DATER [Ye et al., 2023b] guided reasoning through SQL-based parsing and completion. More recently, [Liu et al., 2024a] explored zero-shot prompting with "think step by step" instructions to encourage implicit decomposition.

Agent in TableQA. In the Agent paradigm, LLMs analyze the question, plan steps, and generate Python code to operate over tables [Li et al., 2024a, Gong et al., 2020]. CHAIN-OF-TABLE [Wang et al., 2024] decomposes questions by creating intermediate tables and applying custom functions. ReAcTable [Zhang et al., 2024c] iteratively generates intermediate results and adapts subsequent actions based on output. Other works [Liu et al., 2024b,a] integrate SQL or Python-based agents for structured code-level reasoning.

Joint DP and Agent. DP and Agent can be combined for joint reasoning. Mix-SC [Liu et al., 2024a] merges both paradigms and uses majority voting for answer selection. TIDE [Yang et al., 2025] further introduces structured triplets to enhance decomposition. However, current methods focus on guiding reasoning without addressing token co-occurrence bias from LLM pretraining, which can introduce spurious correlations. To address this, we are the first to define LLM-based TableQA from a causal perspective and identify latent confounding in the reasoning process.

B Front-door Adjustment

Back-door adjustment requires explicit access to the confounding variable C. However, in our setting, the bias induced by LLM pretraining is latent and unobservable, rendering back-door adjustment inapplicable. Fortunately, the front-door adjustment criterion [Pearl et al., 2016] enables causal estimation without requiring access to confounder values. It operates by intervening on the treatment variable using the *do*-operator and leveraging an observed mediator that satisfies the front-door conditions.

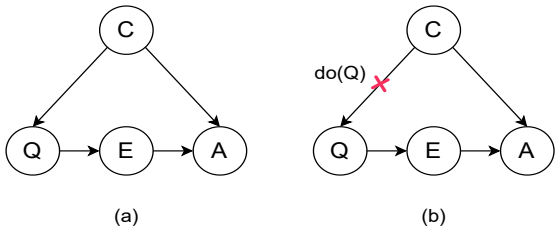

Figure 6: The causal graph of TableQA.

Following the law of total probability, we derive the decomposition expressed in Equation 9.

$$P(A = a \mid do(Q = q)) = \sum_{e} P(A = a \mid do(Q = q), E = e) P(E = e \mid do(Q = q)) \tag{9}$$

By the back-door criterion, the intervention on E does not alter the conditional distribution of A given Q and E. Hence, introducing the *do*-operator on E does not affect the overall expression, yielding Equation 10.

$$P(A = a \mid do(Q = q)) = \sum_{e} P(A = a \mid do(Q = q), do(E = e)) P(E = e \mid do(Q = q)) \tag{10}$$

Under the structural causal model (SCM), since Q and E are connected via a direct causal link without confounders, the *do*-operator on Q can be omitted, yielding Equation 11.

$$P(A = a \mid do(Q = q)) = \sum_{e} P(A = a \mid do(Q = q), do(E = e))P(E = e \mid Q = q) \tag{11}$$

In the SCM, Q and A are not directly connected, so intervening on Q does not affect the distribution of A, yielding Equation 12.

$$P(A = a \mid do(Q = q)) = \sum_{e} P(A = a \mid do(E = e))P(E = e \mid Q = q) \tag{12}$$

Using the law of total probability, we can derive Equation 13.

$$P(A = a \mid do(Q = q)) = \sum_{q'} \sum_{e} P(A = a \mid do(E = e), Q = q')P(Q = q' \mid do(E = e))P(E = e \mid Q = q) \tag{13}$$

Using the same logic as in the transition from Equation 9 to Equation 10, we proceed as Equation 14.

$$P(A = a \mid do(Q = q)) = \sum_{q'} \sum_{e} P(A = a \mid E = e, Q = q')P(Q = q' \mid do(E = e))P(E = e \mid Q = q) \tag{14}$$

Using the same logic as in the transition from Equation 10 to Equation 11, we proceed as Equation 15.

$$P(A = a \mid do(Q = q)) = \sum_{q'} \sum_{e} P(A = a \mid E = e, Q = q')P(Q = q')P(E = e \mid Q = q) \tag{15}$$

Finally, by reorganizing the summation terms, we obtain the Equation 16.

$$P(A = a \mid do(Q = q)) = \sum_{e} P(E = e \mid Q = q) \sum_{q'} P(A = a \mid E = e, Q = q')P(Q = q') \tag{16}$$

C Prompt

We provide the prompts for the three core components in this section.

D Baselines

SASP [Ou and Liu, 2022] uses lexical and structural features to generate programs for solving pseudo programs. TAPAS-large [Eisenschlos et al., 2020] creates a balanced dataset of millions of automatically generated training examples for intermediate learning before fine-tuning. T5-3B [Xie et al., 2022] within the Unified SKG framework unifies 21 SKG tasks into a text-to-text format for comprehensive SKG research. TAPEX-large [Liu et al., 2021] learns a neural SQL executor on a synthetic corpus of executable SQL queries and their outputs. Task Configs [Chen et al., 2023] structured compositional task prompts improve multi-task learning and zero-shot generalization for table-to-text models. TARGET [Ji et al., 2024] is a benchmark for table retrieval in generative tasks, evaluating retriever performance and downstream impacts on QA, fact-checking, and text-to-SQL. TabCot [Chen, 2023b] LLMs excel at table reasoning via chain-of-thought prompting, matching specialized models without table-specific training. TAG-QA [Zhao et al., 2023a] pioneers graph-guided + knowledge-augmented TableQA for long-form answers. UniTabPT[Sarkar and Lausen, 2023] Unified table-pretrained LLMs (T5-based) that outperform specialized models across parsing/QA/classification at scale (770M–11B).

Codex [Cheng et al., 2023], as an OpenAI API, can generate SQL or Python statements and perform end-to-end QA. BINDER [Cheng et al., 2023] combines end-to-end and symbolic approaches,

Instruction: Based on the table content, generate a question similar to the original question without changing its main content, and write it after 'generate questions:'.

Example:

Table:
/*
table caption : stay in office
name	took office	left office	party
William McCreery	March 4, 1803	March 3, 1809	Democratic Republican
Alexander McKim	March 4, 1809	March 3, 1815	Democratic Republican
William Pinkney	March 4, 1815	April 18, 1816	Democratic Republican
Peter Little	September 2, 1816	March 3, 1823	Democratic Republican
Peter Little	March 4, 1823	March 3, 1825	Jacksonian DR
Peter Little	March 4, 1825	March 3, 1829	Adams
Benjamin C. Howard	March 4, 1829	March 3, 1833	Jacksonian
*/
Question: How many people stayed at least 3 years in office?
generate questions:
1. The total number of people stay at least 3 years in office?
2. How many people stayed more than 3 years in office?
3. How many people served in office for 3 years or more?

Test:

Table:
/*
table caption : {TITLE}
{TABLE}
*/
Question: {QUESTION}

Figure 7: The prompt of question variants generation.

Instruction: Analyze the initial questions and the generated similar questions based on the table, find the relevant evidence related to each question, and summarize the questions to obtain the total evidence, and write it after ' evidence:'.

Test:

Table:
/*
table caption : {TITLE}
{TABLE}
*/
Question: {QUESTION}

Figure 8: The prompt of evidence aggregation.

generating and iteratively refining pseudo-SQL queries to construct final answers. For TableQA, DATER [Ye et al., 2023b] extracts relevant sub-tables and decomposes questions to reason jointly over them. StructGPT [Jiang et al., 2023] enhances zero-shot reasoning by iterating through specialized interfaces for structured data. DTE [Wang et al., 2023a] generates counterfactual examples to refine text-to-SQL question answering. TACR [Wu et al., 2023] aligns multi-hop questions with different modalities for accurate evidence retrieval. ITR [Lin et al., 2023] selects relevant rows and columns to form a compact sub-table for efficient reasoning.

[Liu et al., 2024b] creates new tables with external information, enabling SQL queries over both original and new tables to answer. CHAIN-OF-TABLE [Wang et al., 2024] dynamically plans operation chains based on table structure and associated questions. ReAcTable [Zhang et al., 2024c] uses LLMs to iteratively generate intermediate tables, with external code execution for accuracy. Cabinet [Patnaik et al., 2024] removes irrelevant noise in tables to improve LLM reasoning accuracy. Mix-SC [Liu et al., 2024a] explores the combination of CoT and PyAgent to address LLM sensitivity to table structure. SYNTQA [Zhang et al., 2024b] unifies Text-to-SQL (arithmetic/long tables) and

Instruction: the generate questions are similar questions that arise from the question and help you reasoning the original question. Reasoning gives the answer to each generate question step by step, then reasoning the final answer by referring to generate questions and question.
Ensure the final answer format is only "Final Answer: " form, no other form. And ensure the final answer is a number or entity names, as short as possible, without any explanation.

Example:
Table:
/*
constituency number	name	reserved for (sc/st/none)	district	number of electorates (2009)
43	tikamgarh	none	tikamgarh	153,339
44	jatara	sc	tikamgarh	145,555
45	prithvipur	none	tikamgarh	139,110
46	niwari	none	tikamgarh	141,265
47	khargapur	none	tikamgarh	161,546
48	maharajpur	none	chhatarpur	162,460
51	chhatarpur	none	chhatarpur	152,605
52	bijawar	none	chhatarpur	151,159
total : total	total	total	total	1,207,039
*/
Question: Which district has the greatest total number of electorates?
generate questions:
1. Which district has the highest total number of electorates?
2. Which district has the highest electorate count?
3. Which district has the largest number of electorates?
Answer:
1. Sum the number ..... So Tikamgarh has the highest total number of electorates.
2. count the electorate ..... so Tikamgarh has the highest electorate count.
3. Tikamgarh District ....., so answer is Tikamgarh.
Consider the answers above, Tikamgarh has the greatest total number of electorates.
Final Answer: Tikamgarh

Test:
Table:
/*
table caption : {TITLE}
{TABLE}
*/
Question: {QUESTION}
generate questions: {GENERATE}
Answer:

Figure 9: The prompt of answer inference.

E2E TQA (ambiguity/schemas) via answer selection, boosting performance. TIDE [Yang et al., 2025] use structuring triples to help LLMs decompose and validation reasoning context.

E Data Contamination

Mitigating Data Contamination with CIT. Data contamination is a common concern in LLM-based methods, where test samples may appear in the model's training data. To evaluate the robustness of CIT, we compare the performance of direct answering with that of CIT-based reasoning. As shown in Table 8, CIT achieves approximately 27% higher accuracy, indicating that its effectiveness primarily stems from the method itself rather than potential data leakage.

Table 8: Comparison of direct QA for data contamination.

Models	**Accuracy(%)**
Direct QA [Cheng et al., 2023]	48.70
CIT	**76.38**

#	Description	1939/40	1940/41	1941/42	1942/43	1943/44	1944/45	Total
0	Direct War Losses	360,000	NaN	NaN	NaN	NaN	183,000	543,000
1	Murdered	75,000	100,000	116,000	133,000	82,000	NaN	506,000
2	Deaths In Prisons & Camps	69,000	210,000	220,000	266,000	381,000	NaN	1,146,000
3	Deaths Outside of Prisons & Camps	NaN	42,000	71,000	142,000	218,000	NaN	473,000
4	Murdered in Eastern Regions	NaN	NaN	NaN	NaN	NaN	100,000	100,000
5	Deaths in Other Countries	NaN	NaN	NaN	NaN	NaN	NaN	2,000
6	**Total**	504,000	352,000	407,000	541,000	681,000	270,000	2,770,000

Question :	how many people were murdered in 1940/41?
Gold Answer :	100,000
Reason Answer :	100000
Error Analysis :	**Incorrect answer format**
Question :	what is the last description of losses on this chart?
Gold Answer :	Deaths other countries
Reason Answer :	Total
Error Analysis :	**Unable to recognize special line (total)**

Figure 10: Incorrect answer format and unable to recognize special line errors.

Date	Competition	Location	Country	Event	Placing	Nationality
31 October 2008	2008–09 World Cup	Manchester	United Kingdom	Keirin	2	GBR
31 October 2008	2008–09 World Cup	Manchester	United Kingdom	Sprint	1	GBR
1 November 2008	2008–09 World Cup	Manchester	United Kingdom	500 m time trial	1	GBR
1 November 2008	2008–09 World Cup	Manchester	United Kingdom	Sprint	1	GBR
2 November 2008	2008–09 World Cup	Manchester	United Kingdom	Team sprint	1	GBR
2 November 2008	5th International	Manchester	United Kingdom	International keirin	2	GBR
2 November 2008	2008–09 World Cup	Manchester	United Kingdom	Team sprint	1	GBR
2 November 2008	2008–09 World Cup	Manchester	United Kingdom	Keirin	1	GBR
2 November 2008	2008–09 World Cup	Manchester	United Kingdom	Team sprint	1	GBR
13 February 2009	2008–09 World Cup	Copenhagen	Denmark	Team sprint	1	GBR
13 February 2009	2008–09 World Cup	Copenhagen	Denmark	Team sprint	1	GBR
13 February 2009	2008–09 World Cup	Copenhagen	Denmark	Team sprint	1	GBR
13 February 2009	2008–09 World Cup	Copenhagen	Denmark	Sprint	1	GBR
30 October 2009	2009–10 World Cup	Manchester	United Kingdom	Sprint	1	GBR
30 October 2009	2009–10 World Cup	Manchester	United Kingdom	Sprint	1	GBR
30 October 2009	2009–10 World Cup	Manchester	United Kingdom	Keirin	1	GBR
30 October 2009	2009–10 World Cup	Manchester	United Kingdom	500 m time trial	2	GBR
1 November 2009	2009–10 World Cup	Manchester	United Kingdom	Team sprint	1	GBR
1 November 2009	2009–10 World Cup	Manchester	United Kingdom	Team sprint	1	GBR
1 November 2009	2009–10 World Cup	Manchester	United Kingdom	Team sprint	1	GBR

Question :	what is the total number of competition?
Gold Answer :	20
Reason Answer :	unique_competitions = df['Competition'].unique(), Final answer is 3.
Error Analysis :	**Hallucination adds extra conditions (unique)**

Figure 11: Halluciantion adds extra conditions error.

Rank	Cyclist	Team	Time	UCI ProTour Points
1	Alejandro Valverde (ESP)	Caisse d'Epargne	5h 29' 10"	40
2	Alexandr Kolobnev (RUS)	Team CSC Saxo Bank	s.t.	30
3	Davide Rebellin (ITA)	Gerolsteiner	s.t.	25
4	Paolo Bettini (ITA)	Quick Step	s.t.	20
5	Franco Pellizotti (ITA)	Liquigas	s.t.	15
6	Denis Menchov (RUS)	Rabobank	s.t.	11
7	Samuel Sánchez (ESP)	Euskaltel-Euskadi	s.t.	7
8	Stéphane Goubert (FRA)	Ag2r-La Mondiale	+ 2"	5
9	Haimar Zubeldia (ESP)	Euskaltel-Euskadi	+ 2"	3
10	David Moncoutié (FRA)	Cofidis	+ 2"	1

Question :	who was ranked next after davide rebellin?
Gold Answer :	Paolo Bettini (ITA)
Reason Answer :	Paolo Bettini
Error Analysis :	**Incorrect answer format**

Figure 12: Incorrect answer format error.

F Error Case

G Question Types

Question :	**what** country had the most cyclists?
Question Variant :	1. which country had the highest number of cyclists? 2. which nation had the largest representation of cyclists? 3. in which country were the most cyclists from?

Question :	**who** was ranked between denis menchov and stephane goubert?
Question Variant :	1. which cyclist was positioned between denis menchov and stéphane goubert? 2. who ranked just ahead of stéphane goubert and just behind denis menchov? 3. who was the cyclist that came in between denis menchov and stéphane goubert in the rankings?

Question :	**which** year has the most deaths outside of prisons & camps?
Question Variant :	1. in which year were there the highest number of deaths outside of prisons & camps? 2. which year had the greatest number of deaths outside of prisons & camps? 3. in which year were the deaths outside of prisons & camps the highest?

Question :	**where** was the last competition played?
Question Variant :	1. what was the location of the final competition? 2. in which venue did the last competition take place? 3. where did hannes hopley compete last?

Figure 13: Type of wh- questions.

Question :	**how long** did it take the industrial quest to complete the course?
Question Variant :	1. what was the elapsed time for the industrial quest to finish the race? 2. how long did the industrial quest take to complete the sydney to hobart yacht race? 3. what was the total time taken by the industrial quest to finish the course?

Question :	**how many** total films have they already appeared in?
Question Variant :	1. what is the total number of films they have appeared in? 2. how many films has radhika pandit featured in? 3. what is the count of films that radhika pandit has acted in?

Question :	in cycle 4 of austria's next top model, what is the **average** of all the contestants' ages?
Question Variant :	1. what is the average age of all the contestants in cycle 4 of austria's next topmodel? 2. what is the mean age of the contestants in cycle 4 of austria's next topmodel? 3. what is the average age of participants in the fourth cycle of austria's next topmodel?

Figure 14: Type of how- and average questions.