Debiasing Large Language Models via Adaptive Causal Prompting with Sketch-of-Thought

Bowen Li[1], Ziqi Xu[1], Jing Ren[1], Renqiang Luo[2], Xikun Zhang[1], Xiuzhen Zhang[1], Yongli Ren[1], Feng Xia[1]

[1]RMIT University, Australia, [2]Jilin University, China

Correspondence: ziqi.xu@rmit.edu.au

Abstract

Despite notable advancements in prompting methods for Large Language Models (LLMs), such as Chain-of-Thought (CoT), existing strategies still suffer from excessive token usage and limited generalisability across diverse reasoning tasks. To address these limitations, we propose an **A**daptive **C**ausal **P**rompting with **S**ketch-of-Thought (**ACPS**) framework, which leverages structural causal models to infer the causal effect of a query on its answer and adaptively select an appropriate intervention (i.e., standard front-door and conditional front-door adjustments). This design enables generalisable causal reasoning across heterogeneous tasks without task-specific retraining. By replacing verbose CoT with concise Sketch-of-Thought, ACPS enables efficient reasoning that significantly reduces token usage and inference cost. Extensive experiments on multiple reasoning benchmarks and LLMs demonstrate that ACPS consistently outperforms existing prompting baselines in terms of accuracy, robustness, and computational efficiency. The source code can be found at `https://aisuko.github.io/acps/`.

1 Introduction

Large Language Models (LLMs) play a central role in Natural Language Processing (NLP), achieving state-of-the-art results across a wide range of tasks, from open-domain question answering to multi-step logical reasoning (Brown et al., 2020). Building on this success, prompt-based methods extend LLM capabilities without requiring full model retraining. For example, In-Context Learning (ICL) (Xie et al., 2022) introduces example demonstrations directly into the prompt, enabling LLMs to generalise from just a few instances. To support more complex reasoning, Chain-of-Thought (CoT) prompting elicits step-by-step inference, substantially improving performance on multi-hop and logical tasks (Wei et al., 2022).

Despite recent advances, current prompting strategies exhibit two critical shortcomings. First, excessive token generation remains a major issue: CoT prompts often produce unnecessarily lengthy or redundant reasoning chains, with hundreds of tokens per query, which reduces efficiency and increases inference cost (Xu et al., 2025). While methods such as Chain-of-Draft (CoD) (Xu et al., 2025) and Sketch-of-Thought (SoT) (Aytes et al., 2025) attempt to alleviate this, they typically lag behind CoT in accuracy. Second, unfaithful reasoning emerges from internal model biases, where LLMs may rationalise a predisposed answer instead of performing genuine inference. Subtle bias cues in prompts can lead to plausible yet factually incorrect CoTs, resulting in accuracy drops of up to 36% on complex reasoning tasks (Turpin et al., 2023). Such reasoning failures undermine trust in applications that require faithful and verifiable explanations, such as scientific question answering and commonsense inference.

Recent research has explored the integration of causal inference with LLMs as a promising direction for addressing the aforementioned challenges. By incorporating causal principles such as standard front-door adjustment and instrumental variable, it becomes possible to mitigate bias caused by unobserved confounders, which is often interpreted as internal bias in LLMs. These unobserved confounders can introduce spurious correlations between the query and the answer, leading to unfaithful or misleading outputs. For example, Causal Prompting (Zhang et al., 2025) estimates the causal effect of a query its answer by controlling for intermediate reasoning steps such as CoT. Instead of relying on majority voting across multiple reasoning paths, this method selects the answer with the highest estimated causal effect, thereby improving both accuracy and interpretability.

However, despite its potential, existing causality-based prompting methods rely on strong assump-

Findings of the Association for Computational Linguistics: EACL 2026, pages 4481–4499
March 24-29, 2026

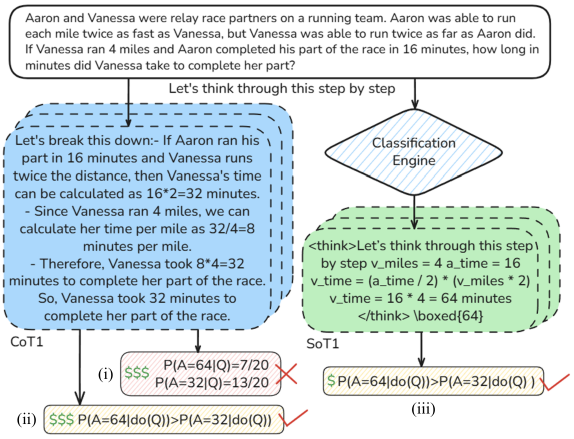


Figure 1: An example from GPT-3.5-turbo on the GSM8K dataset. Left: (i) Recent non-causal prompting methods often amplify internal bias through majority voting. (ii) Some causality-based prompting methods mitigate this bias but rely on verbose CoT, leading to high token usage and inference cost. Right: (iii) The proposed framework uses SoT instead of CoT and selects the answer based on the highest estimated causal effect, yielding the correct result.

tions, including the identifiability conditions required by front-door adjustment and the validity of instrumental variables. These assumptions may not hold consistently across different NLP tasks, limiting the generalisability of such methods in practice. Furthermore, the reliance on CoT often leads to verbose outputs, which increase both token usage and inference cost. Thus, there is a pressing need for a causality-based prompting framework that mitigates internal biases in LLM reasoning by adaptively selecting an appropriate intervention for different NLP tasks, while efficiently reducing token usage without compromising performance.

In this work, we propose an **A**daptive **C**ausal **P**rompting with **S**ketch-of-Thought (**ACPS**) framework to enhance both the generalisability and efficiency of debiasing LLMs. ACPS adaptively applies standard or conditional front-door adjustment depending on the characteristics of each NLP task, supported by a classification engine. To improve inference efficiency, it replaces verbose CoT with concise SoT, significantly reducing token usage and computational cost. As shown in Figure 1, ACPS produces the correct answer on a representative GSM8K example. In contrast, non-causal prompting methods suffer from internal bias, while causality-based prompting methods that rely on verbose CoT are constrained by inefficiency. This highlights how our adaptive formulation and concise reasoning traces contribute to improved accuracy and efficiency in inference. The main contributions of this paper are as follows:

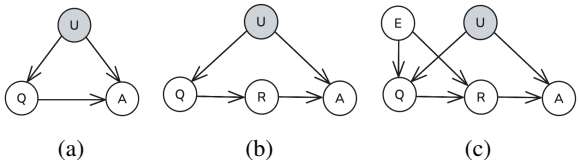
- We propose a novel framework, ACPS, for debiasing LLMs via adaptive causal prompting. This model-agnostic mechanism selects an appropriate intervention based on task characteristics, thereby overcoming the limitations of fixed prompting and improving generalisability across diverse reasoning tasks.
- To the best of our knowledge, ACPS is the first framework that integrates both standard and conditional front-door adjustments with SoT. This integration significantly reduces token usage and inference cost, while preserving high reasoning accuracy.
- We validate our framework through extensive experiments across multiple LLMs and reasoning benchmarks, demonstrating consistent improvements in accuracy, efficiency, and robustness over existing prompting baselines.

2 Preliminaries

In this section, we review the fundamental concepts of causality and sketch-of-thought.

2.1 Structural Causal Model

We adopt the structural causal model (SCM) (Pearl et al., 2016), in which each endogenous variable is determined by a deterministic function of its parent variables and an independent exogenous noise term. The causal structure is represented by a directed acyclic graph (DAG), where nodes correspond to random variables and directed edges denote direct causal relationships. Exogenous variables are assumed to be mutually independent, allowing the joint distribution to factorise according to the graph structure. We assume the Markov condition (Pearl, 2009), which states that each variable is conditionally independent of its non-effects given its direct causes, and the faithfulness assumption (Spirtes et al., 2000), which asserts that all and only those conditional independencies implied by the DAG are present in the observed distribution. Under these assumptions, d-separation (Pearl, 2009) can be used as a criterion to determine conditional independence relationships. Interventions, denoted by $do(X = x)$, modify the structural equation for X, producing a new distribution that reflects the causal



(a) (b) (c)

Figure 2: Three SCMs illustrate different modes of reasoning in LLMs: (a) direct prompt-to-answer reasoning; (b) causality-based prompting for tasks without external knowledge, such as Causal Prompting (Zhang et al., 2025); and (c) causality-based prompting for tasks with external knowledge. Both (b) and (c) are integrated into ACPS. In all SCMs, Q denotes the query, R denotes the reasoning process (SoT or CoT), A denotes the answer, U denotes the unobserved confounder, and E denotes the external knowledge.

effect of the intervention. For formal definition and further details, see (Pearl, 2009).

For direct prompt-to-answer reasoning, the conceptual-level SCM is illustrated in Figure 2a. The query Q, which includes both demonstrations and test examples provided to the LLM, leads to the answer A. Although the direct causal effect from Q to A is represented as $Q \rightarrow A$, LLMs often internalise biases from large-scale pre-training corpora (Ding et al., 2023; Zhao et al., 2025; Ren et al., 2025). To account for these biases, we introduce an unobserved variable U that influences both Q and A. The presence of U creates a spurious association between Q and A, which can result in biased or incorrect answers during inference.

2.2 Sketch-of-Thought

SoT is a prompting framework that rethinks how LLMs express reasoning, addressing the verbosity of CoT (Aytes et al., 2025). Rather than full-sentence explanations, SoT elicits concise sketches, which are abridged representations that capture essential logical structure while omitting details and thereby reducing token usage. The notion of a sketch, drawn from cognitive science (Goel, 1995), denotes a symbolic intermediate form that preserves core reasoning while abstracting away irrelevance. By combining cognitive inspiration with linguistic conciseness, SoT produces compact and interpretable reasoning traces. The SoT template is provided in Appendix H.

3 Methodology

In this section, we first introduce the causal principles, including the standard and conditional front-door criteria. We then describe the mechanism for adaptively selecting an appropriate intervention, followed by the estimation process for each component. Finally, we derive the overall objective function that quantifies the causal effect of the query on its answer. The overall architecture of ACPS is shown in Figure 3.

3.1 Causal Principles

A key approach to handling unobserved confounders is the front-door adjustment (Pearl, 2009). Unlike the back-door criterion, the front-door criterion can isolate causal pathways even when confounders are unobserved. This property makes it particularly suitable for reducing internal biases in LLMs, where unobserved confounders may exist but intermediate variables (i.e., SoT) are observable and controllable. Below, we present the standard front-door criterion and describe its adaptation for prompt-based debiasing.

Definition 1 (Standard Front-Door Criterion) *A set of variables Z_{SFD} is said to satisfy the standard front-door criterion with respect to an ordered variable pair (Q, A) in a DAG $\mathcal{G}$ if the following conditions are met: (1) Z_{SFD} intercepts every directed path from Q to A; (2) there is no unblocked back-door path from Q to Z_{SFD}; (3) all back-door paths from Z_{SFD} to A are blocked by Q.*

The standard front-door criterion offers a theoretical basis for identifying causal effects despite unobserved confounders. In LLMs, this supports using SoT reasoning as a valid front-door variable to estimate the causal effect of prompts on final answers. As shown in Figure 2b, R meets the standard front-door conditions for the causal effect of Q on A. This allows the causal effect $P(A \mid do(Q))$ to be decomposed into two parts: the effect of Q on R, and the effects of R on A conditioned on Q. Formally, the front-door adjustment formula is:

$$P(A \mid do(Q)) = \sum_{r,\, q} \underbrace{P(r \mid Q)}_{①} \underbrace{P(A \mid r, q)}_{②}. \quad (1)$$

However, Figure 2c presents a distinct class of reasoning tasks in which an LLM receives a query Q, generates a SoT R, and produces an answer A. In this setting, external knowledge E influences both Q and R, while an unobserved confounder U introduces spurious correlations that bias causal effect estimation. As this scenario falls outside the scope of the standard front-door criterion, we adopt a conditional front-door adjustment (Xu et al., 2024) to accurately estimate $P(A \mid do(Q))$ and

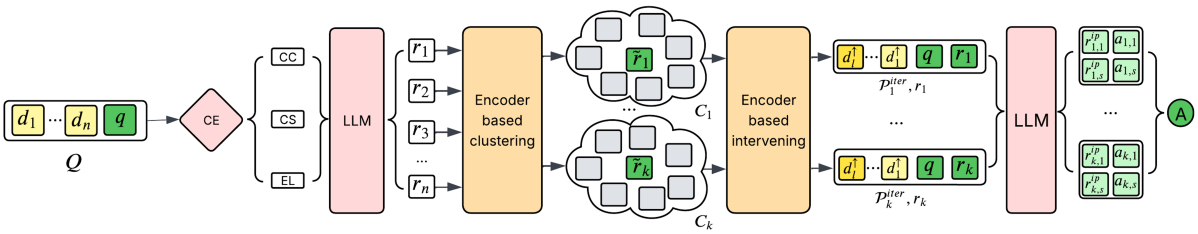


Figure 3: Overall architecture of ACPS. Given an input Q comprising the demonstration examples $[d_1, \ldots, d_n]$ and the test query q, a classification engine (CE) determines an appropriate intervention. The LLM generates M diverse SoTs, which are embedded and clustered into K groups. For each cluster representative, optimal demonstrations are selected via an encoder-based intervention algorithm to form updated prompts $\mathcal{P}_k^{\text{iter}}$. The LLM is then queried S times per prompt, and the final answer is selected as the one associated with the highest estimated causal effect.

identify the most reliable answer. The formal criterion is defined as follows:

Definition 2 (Conditional Front-Door Criterion) *A set of variables Z_{CFD} satisfies the conditional front-door criterion relative to an ordered pair (Q, A) in a DAG $\mathcal{G}$ if the following conditions hold: (1) Z_{CFD} intercepts all directed paths from Q to A; (2) there exists a set of variables W such that all back-door paths from Q to Z_{CFD} are blocked by W; (3) all back-door paths from Z_{CFD} to A are blocked by $Q \cup W$.*

As shown in Figure 2c, R meets all the requirements of the conditional front-door criterion for the pair (Q, A), with the external knowledge variable E acting as the conditioning set W. Thus, R serves as a valid conditional front-door adjustment variable to identify the causal effect of Q and A.

In our framework, Q represents the fixed query during reasoning. Since no intervention is applied to Q, it is considered a constant instead of a random variable. Therefore, the term $P(q \mid e)$ and the summation over q can be removed, simplifying the causal effect expression as follows:

$$P(A \mid do(Q)) = \sum_{r,\, q,\, e} \underbrace{P(r \mid Q, e)}_{①} \underbrace{P(A \mid r, q, e)}_{②} \underbrace{P(e)}_{③} \quad (2)$$

We decompose the causal effect of the query Q on its answer A using two formulations based on the nature of the task. For tasks without external knowledge E, we adopt the standard front-door formulation as shown in Eq. 1, which contains two components that can be independently estimated. For tasks involving external knowledge, we apply the conditional front-door formulation shown in Eq. 2, which extends the decomposition to three components by incorporating conditioning on E. The derivation is provided in Appendix B.

In the following Sections 3.3, 3.4, and 3.5, we detail how each component is estimated in practice. We focus primarily on the conditional front-door setting, as its estimation procedure generalises naturally to the standard front-door case by simply omitting the conditioning on E.

3.2 Classification Engine

To adaptively determine an appropriate intervention for each query, we directly adopt a fine-tuned DistilBERT classification engine (Sanh et al., 2019) from previous work (Aytes et al., 2025) without further training. The engine assigns each query to one of three reasoning paradigms: Conceptual Chaining (CC), Chunked Symbolism (CS), or Expert Lexicons (EL). Given a question x, the classifier outputs logits z_c for each class $c \in \{\text{CC}, \text{CS}, \text{EL}\}$, which are transformed into a probability distribution over the classes using the softmax function:

$$P(c \mid x) = \frac{\exp(z_c)}{\sum_{c' \in C} \exp(z_{c'})}, \quad C = \{\text{CC}, \text{CS}, \text{EL}\}, \quad (3)$$

where $P(c \mid x)$ denotes the probability that the input question x belongs to class c, z_c is the logit associated with class c, and C is the set of candidate reasoning paradigms. The variable c' in the denominator is a dummy index that iterates over all classes in C.

Then, the argmax operation selects the reasoning paradigm with the highest probability:

$$c^* = \arg \max_{c \in \{\text{CC,CS,EL}\}} P(c|x), \quad (4)$$

where c^* denotes the predicted reasoning category with the maximum likelihood.

Each paradigm corresponds to a distinct type of reasoning task. CC includes tasks that require synthesising multiple pieces of contextual or external information. These tasks are handled using

Eq. 2, which applies the conditional front-door adjustment. CS consists of tasks such as arithmetic or algebraic problem solving, where the reasoning involves symbolic steps without reliance on external knowledge. These tasks are addressed using Eq. 1, corresponding to the standard front-door adjustment. EL covers tasks involving commonsense inference and factual verification. For EL, the selection between Eq. 1 and Eq. 2 depends on the availability of external knowledge: if external information is present, the conditional version is used; otherwise, the standard version applies.

3.3 Estimating Reasoning Trace Distribution

We estimate the causal effect between the query Q, the external knowledge E, and the SoT R. To address challenges such as inaccessible LLM output probabilities and limited diversity in SoTs, we generate SoTs by varying the temperature parameter from 0.0 to 2.0 in increments of 0.25, resulting in a diverse set $R = [r_1, r_2, \ldots, r_m]$. To maximise the diversity of the generated SoTs, we compute their embeddings using a pre-trained LLM, Sentence-BERT (Reimers and Gurevych, 2019), as the encoder, producing embeddings $\tilde{R} = [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_m]$. These embeddings are then clustered using the K-Means algorithm (Ikotun et al., 2023) to partition them into K clusters:

$$\{C_1, \ldots, C_k\} = \text{K-means}(\tilde{r}_1, \ldots, \tilde{r}_m), \quad (5)$$

where C_k denotes the k-th cluster in the result, and K is the total number of clusters.

For each cluster, we select the SoT that is closest to the cluster centroid, resulting in a set of K representative SoTs to be used in subsequent causal analysis:

$$r_k = \text{Center}(C_k). \quad (6)$$

We estimate the conditional probability $P(r \mid Q, e)$ based on the size of each cluster as follows:

$$P(r \mid Q, e) \approx \frac{|C_k|}{M}, \quad (7)$$

where $|C_k|$ denotes the number of SoTs in the k-th cluster and M is the total number of generated SoTs.

3.4 Estimating Final Answer Probability

In this section, we estimate the causal effect of the answer A produced by the LLM, conditioned on the query Q, the external knowledge E, and the SoT R. The main challenge arises from the virtually unlimited value space of both Q and E. To address this, we apply the encoder-based Normalised Weighted Geometric Mean (NWGM) approximation (Xu et al., 2015) in ICL prompting interventions. We construct a fixed-size demonstration set $D = \{d_n = (q_n, e_n, r_n^{\text{wrong}}, r_n^{\text{correct}})\}_{n=1}^{N}$, where q_n and e_n represent the question and context of the n-th sample, respectively, and r_n^{wrong} and r_n^{correct} correspond to incorrect and correct SoTs for that sample.

We use the encoder to compute the embedding $\tilde{r}_k$ of the k-th SoT r_k, as selected in Eq. 6 from Section 3.3. Next, we calculate the similarity between $\tilde{r}_k$ and each sample in the ICL demonstration set to improve performance in in-context learning (Margatina et al., 2023). Finally, we rank the ICL demonstrations based on these similarity scores to determine the relevance of each sample, as follows:

$$\{d_n^{\uparrow}\}_{n=1}^{N} = \text{Sort}(D, \tilde{r}_k, \{\tilde{d}_n\}_{n=1}^{N}), \quad (8)$$

where $d_n^{\uparrow}$ denotes the sorted demonstration examples, and $\text{Sort}(\cdot)$ refers to the process of ranking samples using a cosine similarity function $\cos(\cdot, \cdot)$. The demonstrations are ordered such that $\cos(\tilde{r}_k, \tilde{d}_i) \geq \cos(\tilde{r}_k, \tilde{d}_j)$ for all $i < j$.

Then the L most similar samples from the ICL demonstration set are selected to form the prompt, where $L \ll N$. The most similar samples are placed closest to the test query, as this ordering has been shown to better support the encoder-based NWGM algorithm in improving SoT quality through practical experiments. For each SoT r_k of a test sample, the final prompt after intervention is constructed as:

$$\mathcal{P}_k^{\text{iter}} = [d_l^{\uparrow}, \ldots, d_1^{\uparrow}, q^{\text{test}}]. \quad (9)$$

Subsequently, we query the LLM S times using the prompt $\mathcal{P}_k^{\text{iter}}$ and SoT r_k, generating S answers and corresponding improved SoTs. The probability $P(A \mid r, q, e)$ is then estimated as follows:

$$P(A \mid r, q, e) \approx \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}(A = a_{k,s}), \quad (10)$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the generated answer $a_{k,t}$ matches the expected A, and 0 otherwise.

3.5 Estimating External Knowledge Distribution

We maintain the external knowledge E fixed and integrate it directly with the query Q. This integration provides the necessary context for reasoning

and allows the conditional front-door framework to estimate causal effects without explicitly generating or manipulating additional knowledge.

Specifically, each Q is combined with its corresponding E to form a unified input that guides the reasoning process. This joint representation enables the conditional front-door adjustment to capture the underlying causal relationships within multi-hop language processing tasks. By leveraging contextual information directly, the proposed method simplifies the reasoning pipeline while preserving essential dependencies. The distribution of E is assumed to factorise as:

$$P(E) = \prod_i P(e_i), \tag{11}$$

where e_i denotes an individual element within E.

3.6 Objective Function for ACPS

Building on the results from Sections 3.3, 3.4, and 3.5, the final objective can be estimated as follows:

$$\begin{aligned} P(A \mid do(Q)) &= \sum_{r,\, e,\, q} P(r \mid Q, e) \cdot P(A \mid r, q, e) \cdot P(e) \\ &\approx \sum_{k=1}^{K} \left[\prod_{i=1} P(e_i) \right] \cdot \frac{|C_k|}{M} \cdot \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}(A = a_{k,s}) \end{aligned} \tag{12}$$

This equation estimates the causal effect between the query Q and the answer A, enabling the selection of the answer with the highest estimated causal effect as the final unbiased output. The complete learning procedure is detailed in Appendix C.

4 Experiments

4.1 Datasets and Evaluation Setup

Following previous study (Zhang et al., 2025), we evaluate our framework across four categories of reasoning tasks to comprehensively assess its performance: math reasoning (GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021)), commonsense reasoning (CommonsenseQA (ComQA) (Talmor et al., 2019) and StrategyQA (StrQA) (Geva et al., 2021)), multihop reasoning (HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022)), and fact verification (FEVER (Schuster et al., 2019)). Detailed descriptions of the datasets and the evaluation setup are provided in Appendix D.1 and Appendix D.2, respectively.

4.2 Baseline Methods and Backbone LLMs

We evaluate our framework against representative baselines, including ICL (Brown et al., 2020), CoT (Wei et al., 2022), CoT-SC (Wang et al., 2023), SoT (Aytes et al., 2025), CAD (Shi et al., 2024), DeCoT (Wu et al., 2024), and CP (Zhang et al., 2025). Further details are provided in Appendix D.3.

We select three backbone LLMs: Ministral-3B (Mistral AI, 2024), LLaMA-3.1 8B (LLaMA-3) (Grattafiori et al., 2024), and GPT-3.5-turbo (OpenAI, 2022). These models differ in parameter scale and accessibility (open-source versus closed-source), providing a diverse and balanced foundation for comparison in our evaluation.

4.3 Main Results

Table 1 presents the results of ACPS on three backbone LLMs across seven datasets. ACPS consistently achieves the highest or near-highest scores across all benchmarks. In particular, for context-free tasks that do not rely on external knowledge (e.g., GSM8K, MATH, ComQA), ACPS outperforms existing methods by a large margin. For instance, on GSM8K, it improves over CP by 10.04 points on LLaMA-3, indicating its effectiveness even in complex mathematical reasoning. For contextual tasks that require integration of external knowledge (e.g., HotpotQA, MuSiQue, FEVER), ACPS continues to lead, surpassing CAD and DeCoT. On HotpotQA with GPT-3.5-turbo, ACPS obtains 61.31 EM and 75.97 F1, outperforming CP (57.05 EM, 72.51 F1) and DeCoT (53.91 EM, 68.35 F1). Similar gains are observed on MuSiQue and FEVER. These results highlight ACPS's ability to adapt to both types of reasoning by selecting an appropriate intervention and leveraging efficient SoT-based reasoning. Its consistent superiority across datasets and model scales demonstrates the generalisability of the proposed framework.

4.4 Efficiency Analysis

To assess the efficiency of various prompting methods, we conduct a comprehensive analysis across multiple reasoning tasks. Specifically, we measure (i) the average number of reasoning steps taken, (ii) the average number of tokens consumed (in Appendix F.1.2), and (iii) the accuracy-efficiency trade-off under token budgets (in Appendix F.1.3). Figure 4 presents the average number of reasoning steps taken on seven datasets. CoT and CoT-SC often produce lengthy reasoning chains, with more

	Method	GSM8K	MATH	ComQA	StrQA	HotpotQA		MuSiQue		FEVER
		Acc ↑	Acc ↑	Acc ↑	Acc ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	Acc ↑
Ministral-3B	ICL	14.00	4.41	20.00	45.34	26.92	42.06	16.22	31.81	29.41
	CoT	36.09	36.29	44.44	51.72	31.58	47.20	31.05	41.30	31.58
	CoT-SC	42.86	40.12	38.73	55.10	33.33	49.16	40.00	50.02	41.67
	SoT	35.36	36.75	43.45	52.63	31.19	46.37	30.47	40.12	31.15
	CAD	—	—	—	54.24	35.23	49.01	41.25	52.65	39.65
	DeCoT	—	—	—	53.15	35.65	50.21	40.75	53.66	41.58
	CP	**63.16**	33.73	50.00	55.56	38.03	48.98	43.48	51.31	54.55
	ACPS	61.90	**37.93**	**53.67**	**67.80**	**50.00**	**66.67**	**51.72**	**60.65**	**60.00**
LLaMA-3	ICL	18.76	18.75	28.45	56.91	32.55	43.18	32.55	43.18	45.82
	CoT	38.10	40.35	61.73	52.80	39.16	50.20	38.64	48.23	52.45
	CoT-SC	30.77	42.08	57.14	64.29	40.88	55.94	38.46	51.34	53.02
	SoT	37.78	40.83	60.76	52.04	39.17	54.44	42.73	47.24	52.20
	CAD	—	—	—	69.23	53.08	63.95	40.10	53.33	53.55
	DeCoT	—	—	—	70.95	54.63	64.52	41.23	54.56	57.52
	CP	69.67	46.67	60.10	72.10	55.00	69.98	44.94	53.15	63.64
	ACPS	**79.71**	**46.80**	**63.64**	**73.03**	**56.67**	**70.22**	**49.00**	**59.71**	**65.15**
GPT-3.5-turbo	ICL	24.00	20.58	32.69	67.62	46.00	63.72	44.62	57.30	46.15
	CoT	41.79	45.45	65.66	58.83	40.54	58.60	45.94	58.70	47.06
	CoT-SC	62.00	46.33	67.31	72.83	54.74	69.03	47.28	59.90	54.62
	SoT	43.42	46.92	64.25	59.80	40.83	59.04	47.68	60.91	48.21
	CAD	—	—	—	71.79	52.66	67.45	45.05	62.66	64.53
	DeCoT	—	—	—	70.20	53.91	68.35	46.56	63.59	64.69
	CP	**84.18**	**48.36**	**78.03**	73.97	57.05	72.51	75.16	63.61	77.13
	ACPS	81.50	48.33	74.90	**84.45**	**61.31**	**75.97**	**77.01**	**67.15**	**79.48**

Table 1: Performance comparison across seven reasoning datasets. Accuracy (Acc) (%) is reported for GSM8K, MATH, ComQA, StrQA, and FEVER; Exact Match (EM) and F1 scores (%) are reported for HotpotQA and MuSiQue. The best results are shown in **bold**. A dash (–) indicates that the method is not applicable to the dataset, typically because it is designed specifically for knowledge-intensive tasks with external knowledge.

than six steps on datasets such as HotpotQA. In contrast, ACPS consistently generates shorter reasoning traces, typically fewer than three steps, while maintaining strong performance. This demonstrates the superior token efficiency of ACPS compared to more verbose causality-based prompting methods like DeCoT and CP. Further efficiency analyses are provided in Appendix F.1.

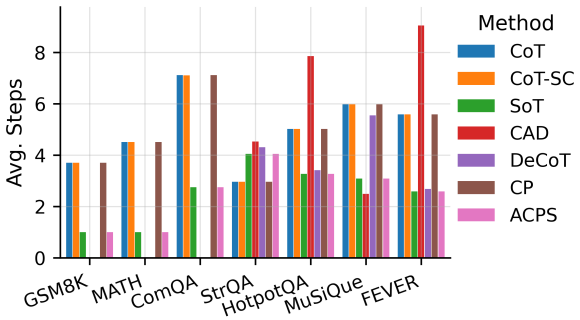


Figure 4: Comparison of the average number of reasoning steps across all datasets for different prompting methods.

4.5 Robustness Study

To assess the robustness of our framework under noisy and disordered scenarios, we conduct experiments on the HotpotQA dataset with two types of data perturbation. In HotpotQA-Inj, one evidence item per record is replaced with unrelated content, introducing semantic noise. In HotpotQA-Shuf, the order of evidence items is shuffled to disrupt contextual flow. These settings evaluate the model's ability to reason under degraded input conditions. As shown in Table 3, ACPS consistently achieves the best performance across the original, injected, and shuffled versions of HotpotQA. In the injected setting, ACPS achieves 3.1% higher EM and 0.8% higher F1 than CP. Under the shuffled setting, ACPS significantly outperforms CP by 8.9% in EM and 8.0% in F1. These results demonstrate that ACPS is robust to input perturbations and effectively handles noisy or disordered inputs through causal SoT-based reasoning.

4.6 Ablation Study

We conduct an ablation study across seven reasoning datasets using GPT-3.5-turbo to evaluate the contribution of key components in ACPS. As shown in Table 2, we compare the full model with several variants: w/o SoT (replacing SoT with CoT), NWGM-Rev (reversing the order of in-context examples), NWGM-Ran (random example selection), w/o K-means (removing clustering), and w/o Weight (omitting causal-effect-based ranking). While the w/o SoT variant achieves the best

	GSM8K	MATH	ComQA	StrQA	HotpotQA		MuSiQue		FEVER
	Acc ↑	Acc ↑	Acc ↑	Acc ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	Acc ↑
ACPS	**81.50**	48.33	74.90	**84.45**	**61.31**	**75.97**	**77.01**	**67.15**	79.48
w/o SoT	80.75	**52.95**	**77.11**	83.17	60.15	75.26	75.15	66.45	**83.20**
NWGM-Rev	81.47	48.05	74.21	84.05	60.67	74.66	76.89	66.67	78.89
NWGM-Ran	81.25	46.67	73.89	82.63	59.21	73.41	75.33	64.01	74.40
w/o K-means	80.95	44.63	72.25	81.50	57.32	68.92	74.13	62.30	73.15
w/o Weight	78.57	41.75	71.43	77.80	55.87	66.97	62.45	60.71	71.41

Table 2: Ablation study results on seven datasets using GPT-3.5-turbo. The best results are shown in **bold**.

	HotpotQA		HotpotQA-Inj		HotpotQA-Shuf	
Method	EM ↑	F1 ↑	EM ↑	F1 ↑	EM ↑	F1 ↑
ICL	46.00	63.72	31.05	42.30	52.89	66.99
CoT	40.54	58.60	31.35	42.28	52.77	67.45
CoT-SC	54.74	69.03	28.57	40.18	51.97	66.31
SoT	40.83	59.04	32.01	43.85	53.15	66.55
CAD	52.66	67.45	29.11	42.40	52.41	66.59
DeCoT	53.91	68.35	29.47	26.89	51.85	67.43
CP	57.05	72.51	34.55	47.97	51.27	67.87
ACPS	**61.31**	**75.97**	**37.68**	**48.78**	**60.20**	**75.84**

Table 3: Robustness results on the HotpotQA dataset using GPT-3.5-turbo. The best results are shown in **bold**.

results on MATH, ComQA, and FEVER, SoT offers consistent advantages on most datasets and remains competitive overall, highlighting its value as a concise yet effective reasoning paradigm. Other ablations cause notable drops, with the largest from w/o Weight (e.g., MuSiQue F1: 67.15 → 60.71), confirming the importance of causal-effect-based selection. Removing K-means also reduces performance (e.g., HotpotQA F1: 75.97 → 68.92), showing the benefit of reasoning diversity. Moreover, NWGM-Ran underperforms NWGM-Rev, indicating that example relevance matters more than order. Overall, these results demonstrate that both similarity-guided prompt construction and causal-effect-aware selection are essential for robust reasoning in ACPS.

4.7 Hyper-parameter Study

We conduct a hyper-parameter study on the number of generated SoTs (M) and clusters (K). We observe that larger values generally improve performance but incur higher computational cost. The complete results are provided in Appendix F.2.

4.8 Additional Details

Due to page limitations, further implementation details, including the core component setup and demonstration step, are provided in Appendix E. The prompting template is given in Appendix H, and case studies are presented in Appendix I.

5 Related Work

Mitigating biases in LLMs increasingly relies on causal inference frameworks (Pearl, 2009; Pearl et al., 2016; Xu et al., 2023). With strong theoretical guarantees, various methods have been developed to estimate causal effects even in the presence of unobserved confounders (Cheng et al., 2024a,b; Du et al., 2025). As a result, LLM reasoning has been increasingly framed within SCMs to reduce spurious correlations. For example, DeCoT applies front-door adjustment using CoT as a mediator and external knowledge as an instrumental variable, improving reasoning performance but limiting generality due to its reliance on external inputs (Wu et al., 2024). Similarly, Causal Prompting employs CoT-based front-door adjustment enhanced by contrastive learning; however, its causal guarantees do not generalise to all task types (Zhang et al., 2025). See Appendix G for more related work.

Our framework differs from prior methods by unifying the standard and conditional front-door adjustments within a single framework, addressing both context-free and context-dependent reasoning tasks. We further incorporate SoT to improve inference efficiency. This design enhances scalability and generality, offering an effective and principled solution for mitigating bias in LLMs.

6 Conclusion

In this paper, we present ACPS, a novel prompting framework that integrates standard and conditional front-door adjustments with efficient Sketch-of-Thought reasoning. By adaptively selecting the appropriate intervention based on task characteristics, ACPS mitigates internal biases in large language models while substantially reducing token usage and inference cost. Extensive experiments demonstrate that ACPS consistently outperforms existing prompting baselines in accuracy, efficiency, and robustness.

7 Limitations

Although our results demonstrate the effectiveness of our framework, several aspects warrant further exploration. Expanding the evaluation to larger-scale test sets and more powerful backbone models could provide deeper insights into the generalizability and scalability of the framework. In addition, while we vary the temperature to encourage diversity in SoTs, certain edge-case generations remain constrained by the API's safety policy; future work may consider strategies to address such cases while remaining compliant with safety requirements. Finally, further validation on broader datasets and in real-world scenarios would help strengthen the evidence of robustness and practical applicability.

References

Simon A. Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient LLM reasoning with adaptive cognitive-inspired sketching. *CoRR*, abs/2503.05179.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*.

Debo Cheng, Jiuyong Li, Lin Liu, Ziqi Xu, Weijia Zhang, Jixue Liu, and Thuc Duy Le. 2024a. Disentangled representation learning for causal inference with instruments. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14.

Debo Cheng, Ziqi Xu, Jiuyong Li, Lin Liu, Jixue Liu, Wentao Gao, and Thuc Duy Le. 2024b. Instrumental variable estimation for causal inference in longitudinal data with time-dependent latent confounders. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI*, pages 11480–11488.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Xiaojing Du, Jiuyong Li, Debo Cheng, Lin Liu, Wentao Gao, Xiongren Chen, and Ziqi Xu. 2025. Telling peer direct effects from indirect effects in observational network data. In *Forty-second International Conference on Machine Learning, ICML*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Vinod Goel. 1995. *Sketches of Thought*. MIT Press, Cambridge, MA.

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 10 others. 2024. The llama 3 herd of models. Accessed: 2025-07-15.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks*.

Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, and Heming Jia. 2023. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. 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, IJCNLP*, pages 305–329.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 5011–5034.

Mistral AI. 2024. Ministral 3b instruct. Open-source language model. Accessed: 2025-07-15.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`. Accessed: 2025-07-24.

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

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jing Ren, Wenhao Zhou, Bowen Li, Mujie Liu, Nguyen Linh Dan Le, Jiade Cen, Liping Chen, Ziqi Xu, Xiwei Xu, and Xiaodong Li. 2025. Causal prompting for implicit sentiment analysis with large language models. *arXiv preprint arXiv:2507.00389*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Tal Schuster, Darsh J. Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 3417–3423.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Short Papers, NAACL*, pages 783–791.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. 2000. *Causation, prediction, and search*. MIT Press.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4149–4158.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS*.

Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan A. Rossi, Sungchul Kim, Anup B. Rao, and Julian J. McAuley. 2024. Decot: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*, pages 14073–14087.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, volume 37, pages 2048–2057.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *CoRR*, abs/2502.18600.

Ziqi Xu, Debo Cheng, Jiuyong Li, Jixue Liu, Lin Liu, and Ke Wang. 2023. Disentangled representation for causal mediation analysis. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*, pages 10666–10674.

Ziqi Xu, Debo Cheng, Jiuyong Li, Jixue Liu, Lin Liu, and Kui Yu. 2024. Causal inference with conditional front-door adjustment and identifiable variational autoencoder. In *The Twelfth International Conference on Learning Representations, ICLR*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Congzhi Zhang, Linhai Zhang, Jialong Wu, Yulan He, and Deyu Zhou. 2025. Causal prompting: Debiasing large language model prompting based on front-door adjustment. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, AAAI* , pages 25842–25850.

Bo Zhao, Yinghao Zhang, Ziqi Xu, Yongli Ren, Xiuzhen Zhang, Renqiang Luo, Zaiwen Feng, and Feng Xia. 2025. Unbiased reasoning for knowledge-intensive tasks in large language models via conditional front-door adjustment. In *Proceedings of the 34th ACM*

International Conference on Information and Knowledge Management, CIKM, pages 4315–4325.

A Discussion

A.1 Why a single unobserved confounder U?

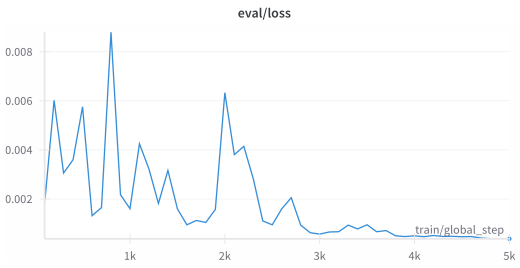
Although real-world reasoning may involve multiple unobserved confounders, our assumption of a single unobserved confounder in the SCM is consistent with prior work and facilitates tractable causal analysis via the front-door criterion. Empirically, our experiments show that LLMs often initiate reasoning correctly but tend to fail at the final step due to internal biases. This pattern indicates the presence of a dominant confounder that substantially influences the direct relationship between the query and the answer, thereby supporting the validity of our simplified SCM. Moreover, the ACPS framework addresses potential biases introduced by reasoning traces by generating diverse SoTs through clustering and applying the NWGM algorithm for prompt selection. Consequently, the single unobserved confounder assumption proves to be both practically robust and computationally efficient. Future work will consider more complex causal structures to model additional confounding factors more comprehensively.

A.2 Why not fine-tune the encoder?

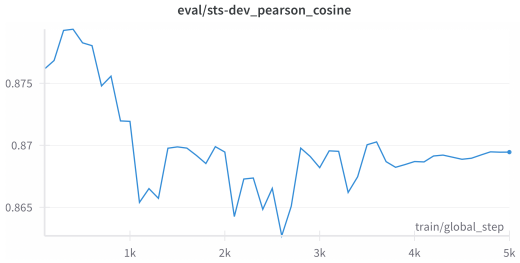
We fine-tune the encoder using the `SentenceTransformerTrainer` (Reimers and Gurevych, 2019) on GSM8K (4,096 examples) with 20 epochs, a batch size of 16, and a learning rate of 1×10^{-4}. However, the training shows unstable evaluation loss and highly variable correlation metrics (`sts-dev_pearson_cosine`) (see Figure 5). In addition, a persistent gap between training and evaluation loss indicates rapid overfitting, suggesting that the dataset is too small to support reliable fine-tuning. To avoid these issues, we use the pre-trained `Sentence-BERT` encoder without additional fine-tuning, which provides more stable embeddings and better generalisation across tasks.

A.3 What happens if classification fails?

When the classification engine fails to produce a reliable output, we employ a default template grounded in commonsense knowledge (CS). This template is applicable to both contextualised and uncontextualised tasks, thereby ensuring robust and consistent performance across all task types.



(a)



(b)

Figure 5: Trends of (a) evaluation loss and (b) `sts-dev_pearson_cosine` during encoder fine-tuning on GSM8K.

B Detailed Derivations

Theorem 1 (Rules of do-Calculus) *(Pearl, 2009) Let $\mathcal{G}$ be the DAG associated with a structural causal model, and let $P(\cdot)$ denote the probability distribution induced by that model. For any disjoint subsets of variables Q, A, Z, and W, the following rules hold:*

- *Rule 1. (Insertion/deletion of observations):*

$$P(A \mid do(Q), Z, W) = P(A \mid do(Q), W) \quad \text{if } (A \perp\!\!\!\perp Z \mid Q, W) \text{ in } \mathcal{G}_{\overline{Q}}.$$

- *Rule 2. (Action/observation exchange):*

$$P(A \mid do(Q), do(Z), W) = P(A \mid do(Q), Z, W) \quad \text{if } (Y \perp\!\!\!\perp Z \mid Q, W) \text{ in } \mathcal{G}_{\overline{Q}\underline{Z}}.$$

- *Rule 3. (Insertion/deletion of actions):*

$$P(A \mid do(Q), do(Z), W) = P(A \mid do(Q), W) \quad \text{if } (A \perp\!\!\!\perp Z \mid Q, W) \text{ in } \mathcal{G}_{\overline{Q},\overline{Z(W)}}.$$

where $Z(W)$ is the set of nodes in Z that are not ancestors of any node in W in $\mathcal{G}_{\overline{Q}}$.

As shown in Figure 2c, R meets all the requirements of the conditional front-door criterion for the pair (Q, A), with the external knowledge variable E acting as the conditioning set W. Thus, R serves as a valid conditional front-door adjustment variable to identify the causal effect of Q and A. We now apply Theorem 2 to to derive $P(A \mid do(Q))$, with the derivation process detailed below:

$$
\begin{aligned}
P(A \mid do(Q)) &= \sum_r P(r \mid do(Q)) P(A \mid r, do(Q)) \\
&= \sum_r P(r \mid do(Q)) \sum_e P(A \mid \mathrm{do}(Q), r, e) P(e \mid \mathrm{do}(Q), r) \\
&= \sum_r P(r \mid do(Q)) \sum_e P(A \mid \mathrm{do}(Q), \mathrm{do}(r), e) P(e \mid \mathrm{do}(Q), r), \\
&\quad \text{since } (A \perp\!\!\!\perp R \mid Q, E) \text{ in } \mathcal{G}_{\overline{Q}\underline{R}} \text{ (Rule 2 in Theorem 1)} \\
&= \sum_r P(r \mid do(Q)) \sum_e P(A \mid \mathrm{do}(r), e) P(e \mid \mathrm{do}(Q), r), \\
&\quad \text{since } (A \perp\!\!\!\perp Q \mid R, E) \text{ in } \mathcal{G}_{\overline{RQ(E)}} \text{ (Rule 3 in Theorem 1)} \\
&= \sum_r P(r \mid do(Q)) \sum_{e,\,q} P(A \mid \mathrm{do}(r), q, e) P(q \mid \mathrm{do}(r), e) P(e \mid \mathrm{do}(Q), r), \\
&= \sum_r P(r \mid do(Q)) \sum_{e,\,q} P(A \mid r, q, e) P(q \mid \mathrm{do}(r), e) P(e \mid \mathrm{do}(Q), r), \\
&\quad \text{since } (A \perp\!\!\!\perp R \mid Q, E) \text{ in } \mathcal{G}_{\underline{R}} \text{ (Rule 2 in Theorem 1)} \\
&= \sum_r P(r \mid do(Q)) \sum_{e,\,q} P(A \mid r, q, e) P(q \mid e) P(e \mid \mathrm{do}(Q), r), \\
&\quad \text{since } (Q \perp\!\!\!\perp R \mid E) \text{ in } \mathcal{G}_{\overline{R(E)}} \text{ (Rule 3 in Theorem 1)} \\
&= \sum_r P(r \mid do(Q)) \sum_{e,\,q} P(A \mid r, q, e) P(q \mid e) \frac{P(e, r \mid do(Q))}{P(r \mid \mathrm{do}(Q))}, \\
&\quad \text{since the chain rule of conditional probability} \\
&= \sum_r P(r \mid do(Q)) \sum_{e,\,q} P(A \mid r, q, e) P(q \mid e) \frac{P(r \mid Q, e) p(e)}{P(r \mid \mathrm{do}(Q))} \\
&= \sum_{r,\,e} P(r \mid Q, e) \sum_{q,\,e} P(A \mid r, q, e) P(q \mid e) P(e)
\end{aligned}
$$

In our framework, Q represents the fixed input query during reasoning. Since no intervention is applied to Q, it is considered a constant instead of a random variable. Therefore, the term $P(q \mid e)$ and the summation over q can be removed, simplifying the causal effect expression as follows:

$$
P(A \mid do(Q)) = \sum_{r,\,q,\,e} \underbrace{P(r \mid Q, e)}_{①} \underbrace{P(A \mid r, q, e)}_{②} \underbrace{P(e)}_{③}
$$

C ACPS Algorithm

We describe the ACPS procedure in detail in Algorithm 1.

Algorithm 1 ACPS

Input: Query Q, External knowledge E, Encoder, D, d, LLM
Parameters: M (number of initial SoTs), K (number of clusters),

$prompt \leftarrow [d_1, ...d_n, \{Q, E\}]$
$R_{\text{init}} = [r_1, r_2, \ldots, r_m] \leftarrow \text{LLM}(prompt)$
$\tilde{R}_{\text{init}} \leftarrow \text{Encoder}(R_{\text{init}})$
$\tilde{R} = [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_k] \leftarrow \text{K-means}(\tilde{R}_{\text{init}})$ $n = 1$ to N
Compute $P(r_k \mid Q, e)$ using Eq. 6 and 7
Compute $P(A \mid r_k, Q, e)$ using Eq. 8, 9 and 10
Compute $P(A \mid do(Q))$ using Eq. 12
$\arg\max_A P(A = a \mid do(Q))$

D Experimental Details

D.1 Dataset Details

- **GSM8K** (Cobbe et al., 2021) comprises 8.5K grade-school mathematics word problems requiring 2–8 arithmetic steps.
- **MATH** (Hendrycks et al., 2021) contains 12.5K competition-level mathematics problems with detailed step-by-step solutions.
- **CommonsenseQA** (Talmor et al., 2019) includes 12,247 multiple-choice questions grounded in ConceptNet relations, designed to probe background world knowledge.
- **StrategyQA** (Geva et al., 2021) is a yes/no question answering benchmark comprising 2,780 questions that require implicit multi-step reasoning. Each question is annotated with a decomposition and supporting Wikipedia paragraphs.
- **HotpotQA** (Yang et al., 2018) consists of 113K Wikipedia-based question–answer pairs that require multi-hop reasoning across documents.
- **MuSiQue** (Trivedi et al., 2022) contains approximately 25K two- to four-hop questions constructed by composing single-hop questions from existing datasets.
- **Symmetric FEVER** (Schuster et al., 2019) is a diagnostically enhanced evaluation set derived from the original FEVER benchmark, comprising 1,420 counterfactual claim–evidence pairs. Each instance contains one supporting and one refuting Wikipedia sentence, labelled as SUPPORT or REFUTE, specifically constructed to expose and mitigate claim-only biases in fact-verification systems. For brevity, we refer to Symmetric FEVER as "FEVER" throughout the remainder of this paper.

All datasets used in this work (GSM8K, MATH, CommonsenseQA, StrategyQA, Hot-

Dataset	Measure / Eval Type
GSM8K	Accuracy / numerical
MATH	Accuracy / open
CommonsenseQA	Accuracy / multiple_choice
StrategyQA	Accuracy / yes/no
HotpotQA	F1 & EM / open
MuSiQue	F1 & EM / open
FEVER	Accuracy / supports/refutes

Table 4: Details of dataset setup for experiments.

potQA, MuSiQue, and FEVER) are publicly available under their respective research licenses and are used strictly for research purposes, consistent with their intended use. These benchmarks do not contain personally identifying information. While some datasets may include open-domain text with potentially sensitive content, they have been widely adopted in prior work and released with appropriate safeguards. We did not collect new data, and no additional anonymization was necessary.

D.2 Evaluation Setup

Consistent with prior work (Lyu et al., 2023; Zhang et al., 2025), we evaluate the performance of ACPS across different reasoning paradigms. For Chunked Symbolism tasks, we use label classification accuracy (Acc), which is appropriate for mathematical reasoning involving numerical and symbolic operations. For Conceptual Chaining tasks, including CommonsenseQA and StrategyQA, we also use accuracy, given the nature of these tasks, which require connecting ideas in logical sequences. For Multihop Reasoning tasks, we adopt Exact Match (EM) and F1 scores, as these metrics are better suited for problems that require multi-step reasoning across multiple pieces of information. In line with Aytes et al. (2025), we extract the answer text span enclosed within the `\boxed{}` keyword when evaluating span-based reasoning tasks. The dataset setup details are provided in Table 4.

D.3 Baseline Methods

We briefly describe the baseline methods considered in our evaluation:

- **In-Context Learning (ICL)** (Brown et al., 2020): Uses input–output demonstrations without explicit reasoning steps.
- **Chain-of-Thought (CoT)** (Wei et al., 2022): Incorporates intermediate reasoning steps to support logical inference.
- **CoT with Self-Consistency (CoT-SC)** (Wang et al., 2023): Samples multiple reasoning paths and selects the majority answer.
- **Sketch-of-Thought (SoT)** (Aytes et al., 2025): Produces concise reasoning sketches that capture essential logic while reducing token usage.
- **Context-Aware Decoding (CAD)** (Shi et al., 2024): Enhances reliability by comparing model outputs generated with and without additional context.
- **Debiasing CoT (DeCoT)** (Wu et al., 2024): Mitigates bias via front-door adjustment with external knowledge.
- **Causal Prompting (CP)** (Zhang et al., 2025): Estimates causal effects between prompts and answers using front-door adjustment.

E Implementation Details

E.1 LLM Setup

We design a custom asynchronous client for the Microsoft Azure serverless inference API to support concurrent requests during experimentation.

E.2 Classification Engine Setup

In line with (Aytes et al., 2025), we employ the router model as our classification engine. This model assigns queries to distinct reasoning paradigms and is applied without further training or modification, maintaining consistency with established reasoning frameworks.

E.3 Encoder Setup

We use Sentence-BERT (Reimers and Gurevych, 2019), a pre-trained language model, as the encoder for computing sentence embeddings. The embeddings are then used for similarity measurement, SoT clustering, and in-context demonstration selection within the NWGM algorithm.

E.4 Demonstration Construction

We standardise the demonstration construction process across all datasets by designating the answer column (i.e., the ground truth) as the reference label. Rather than relying on gold rationales or manual annotations, we automatically generate both correct and incorrect SoTs by sampling model outputs under different temperature settings.

Higher temperatures increase diversity and creativity, yielding a mix of accurate and flawed reasoning paths. For the demonstration set, we randomly select questions associated with both correct and incorrect SoTs to capture diverse reasoning patterns. To rigorously evaluate the debiasing effect, demonstrations are constructed only from the original dataset, while evaluation is performed on both the original and adversarial datasets. This design ensures a consistent and scalable demonstration pipeline applicable across multiple tasks.

E.5 Demonstration Selection

We select the most relevant demonstrations for each query based on embedding similarity and concatenate them into the prompt. For each dataset, a dedicated demonstration set is constructed. By default, we use two demonstrations per task ($l = 2$ in Eq. 9), ensuring a consistent few-shot prompting configuration across datasets. This design keeps the selection and integration process uniform and reproducible across all tasks in our study.

E.6 SoT Generation and Answer Selection

To minimise computational overhead, we pre-generate all SoTs for each query before embedding computation. We sample SoTs with temperatures ranging from 0.0 to 2.0 in increments of 0.25, while fixing top_p at 0.9 to encourage diversity. This yields $M = 9$ SoTs per query, which are clustered into $K = 4$ groups using K-Means. For each cluster centroid, we generate $S = 3$ answers via prompts refined through our causal intervention procedure. The resulting $K \times S = 12$ answers are then aggregated by causality-weighted voting to obtain the final prediction.

F Experimental Results

F.1 Efficient Analysis

F.1.1 Efficiency Comparison of CoT and SoT

We evaluate the efficiency of CoT and SoT by comparing the number of tokens and reasoning steps on identical questions across tasks. Figures 6 and 7 show that SoT achieves comparable performance with fewer tokens and shorter reasoning paths, highlighting its superior efficiency.

F.1.2 Token Consumption Analysis

We analyse and compare the average tokens consumed by different prompting frameworks across multiple tasks, as shown in Figure 8. The results

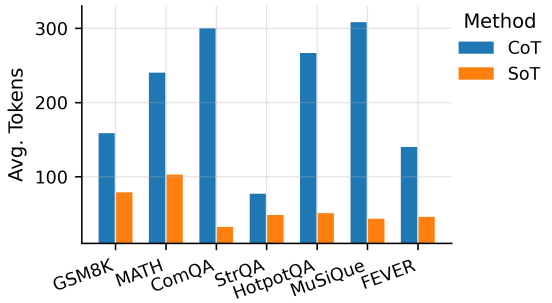


Figure 6: Comparison of average tokens consumed between CoT and SoT.

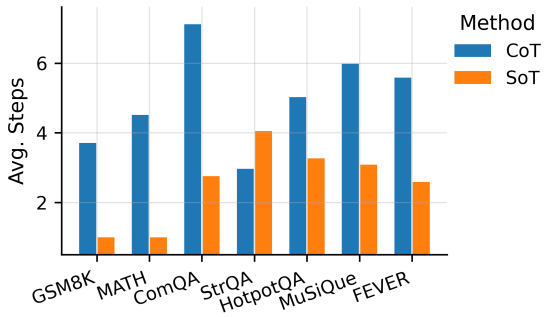


Figure 7: Comparison of average reasoning steps between CoT and SoT.

demonstrate that ACPS consistently requires fewer tokens while maintaining competitive performance, thereby demonstrating superior efficiency across diverse reasoning benchmarks.

F.1.3 Accuracy-Efficiency Trade-off under Token Budgets

To examine the accuracy-efficiency trade-off under varying token budgets, we compare CP, ACPS, and ACPS-CoT on GPT-3.5-turbo using the StrategyQA and HotpotQA datasets, with `max_tokens` varied up to 500. As shown in Figures 9 and 10, ACPS consistently achieves the highest accuracy, demonstrating superior token efficiency. ACPS-CoT performs between CP and ACPS, yielding slightly better results than CP under the same token budgets but still falling short of ACPS. At comparable performance levels, ACPS requires fewer tokens than both CP and ACPS-CoT, further confirming its efficiency advantage.

F.2 Hyper-parameter Study

We conduct a hyper-parameter study to examine how varying the number of initially generated SoTs (M) and the number of clusters (K) influences the performance of our framework. Due to computational constraints, we explore $M \in \{4, 6, 8, 10, 12\}$ and $N \in \{1, 3, 5, 7, 9\}$. The re-

	GSM8K	MATH	ComQA	StrQA	HotpotQA		MuSiQue		FEVER
M	Acc ↑	Acc ↑	Acc ↑	Acc ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	Acc ↑
4	78.58	47.14	74.09	80.04	59.12	73.37	75.67	65.11	75.15
6	79.00	47.89	74.55	81.01	60.41	75.44	75.84	65.55	76.64
8	79.25	48.21	74.75	83.06	61.11	75.59	76.21	66.92	78.23
10	81.52	48.36	74.95	84.12	62.23	75.49	77.03	67.21	79.84
12	82.30	47.65	75.10	84.32	61.56	75.69	77.89	67.94	80.11
K	Acc ↑	Acc ↑	Acc ↑	Acc ↑	EM ↑	F1 ↑	EM ↑	F1 ↑	Acc ↑
1	77.25	47.25	72.84	78.01	60.25	76.67	75.59	66.01	76.89
3	78.92	49.51	73.21	84.20	61.11	77.88	76.52	66.11	77.24
5	84.18	51.10	75.34	81.11	61.33	76.73	77.12	68.19	78.13
7	80.00	52.12	75.21	80.20	59.65	76.23	77.81	69.10	78.87
9	81.50	53.21	76.80	75.03	58.22	75.03	77.95	69.12	79.32

Table 5: The performance of ACPS under different numbers of generated SoTs (M) and clusters (K) across seven datasets.

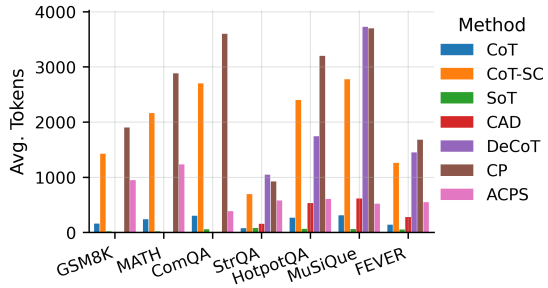


Figure 8: Comparison of average token consumed across all datasets for different prompting methods.

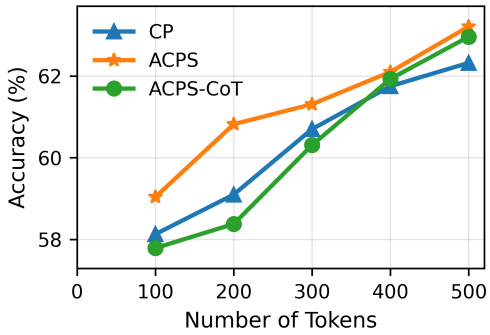


Figure 9: Comparison among CP, ACPS-CoT, and ACPS under varying token budgets on GPT-3.5-turbo for the HotpotQA dataset.

sults are reported in Table 5. Overall, increasing M generally improves performance, as a larger and more diverse set of initial SoTs captures richer reasoning trajectories. Similarly, increasing N allows for finer-grained clustering of SoTs, which enhances the robustness of causal effect estimation. However, both higher M and N values lead to greater token consumption and computational overhead.

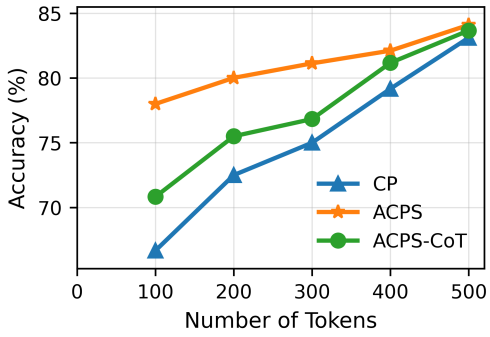


Figure 10: Comparison among CP, ACPS-CoT, and ACPS under varying token budgets on GPT-3.5-turbo for the StrategyQA dataset.

G Related Work

Large language models have shown impressive performance on various NLP tasks when provided with effective prompts. To avoid the high costs of scaling model size, researchers have developed prompt-based strategies that enhance reasoning without additional training. ICL enables models to learn from a few examples within the prompt (Brown et al., 2020), while CoT prompting encourages step-by-step reasoning to improve multi-hop inference (Wei et al., 2022). To address answer variability, self-consistency decoding samples multiple reasoning paths and selects the majority answer (Wang et al., 2023). More recently, SoT generates concise reasoning sketches to improve efficiency across tasks (Aytes et al., 2025).

Despite their effectiveness, these strategies primarily rely on correlational signals, i.e., selecting examples or reasoning paths based on majority voting. Such reliance can reinforce internal biases within LLMs, leading to unfaithful outputs. This

highlights the need for causally grounded prompting strategies that can more reliably guide model reasoning.

H Prompt Templates

This section presents the prompt templates for SoT prompting, along with those used in conjunction with the NWGM approximation.

H.1 SoT Prompting

Chunked Symbolism Prompt Template

Instruction
You are a reasoning expert specializing in **Chunked Symbolism**, a cognitive reasoning technique that organizes numerical reasoning into structured steps. Your goal is to utilize chunked symbolism by representing information through **equations, variables, and step-by-step arithmetic**, using minimal words. Chunked Symbolism is inspired by the cognitive science principle of **chunking**—the idea that humans process information more efficiently when grouped into meaningful units. Rather than solving problems in a free-form manner, Chunked Symbolism breaks down complex operations into smaller, structured steps.
This method is particularly effective for:

- Mathematical problems (arithmetic, algebra, physics, engineering)
- Symbolic reasoning (logic-based computations, formula derivations)
- Technical calculations (financial modeling, physics simulations, unit conversions)

How to Apply Chunked Symbolism:

1. Identify variables—extract relevant numerical values and define variables.
2. Write equations—represent the solution using explicit mathematical formulas.
3. Perform step-by-step computations—solve in small, logical steps, keeping each line clear.
4. Label units—maintain consistent unit representation to prevent ambiguity.
5. Final answer formatting—present the answer in the provided format for clarity.

Rules & Directives:

- Use equations & variables; define variables before computation and always use explicit equations to represent reasoning.
- Avoid redundant text; do not restate the problem—go directly to calculations, and use minimal context only if it aids understanding.
- Apply step-by-step arithmetic; break operations into small, structured steps and ensure each line contains only one computation for clarity.
- **Output format:**

```
<think>
Let's think through this step
by step
[stepwise           equations,
variables, and computations]
</think>
\boxed{[Final answer]}
```

- For multiple-choice, return the correct letter option in the box.
  Always use minimal words.

Demonstration

1. Q: The question is: [question]
2. Let us think step by step.
3. A: `<think>`
   `Let's think through this step by step`
   `[equations, variables, computations]`
   `</think>`
   `\boxed{answer}`

Test example:

1. Q: The question is: [question]
2. Let us think step by step.
3. A: `<think>`
   `Let's think through this step by step`
   `[equations, variables, computations]`
   `</think>`
   `\boxed{answer}`

Structured Concept Linking Prompt Template

Instruction
You are a reasoning expert specializing in **structured concept linking** by connecting essential ideas in a logical sequence. Your goal is to **extract key terms** and present reasoning in **clear, stepwise chains** with minimal explanation.
This method integrates **associative recall** (direct lookups) and **multi-hop reasoning** (sequential dependencies) into a unified framework.
This method is most effective for:

- Commonsense reasoning (linking familiar ideas)
- Multi-hop inference (tracing logical or causal dependencies)
- Fact-based recall (retrieving knowledge with minimal cognitive load)

How to Apply:

1. Extract key concepts—identify the most relevant words or entities.
2. Use minimal words—make each reasoning step concise and direct.
3. Link steps sequentially—ensure clear and meaningful progression between concepts.
4. Avoid full sentences—respond using structured keyword connections.
5. Follow the required format—present answers as stepwise chains.

Rules & Directives:

- Use structured concept linking; each step must be logically connected (arrows (→) for dependencies).
- Avoid unnecessary text; do not restate the question or use full sentences.
- Maintain logical flow; concepts must be meaningfully ordered and contribute to the reasoning process.
- **Output format:**

```
<think>
Let's think through this step
by step
[shorthand reasoning]
</think>
\boxed{[Final answer]}
```

- For multiple-choice, return the correct letter option in the box.
  For fact-based recall, return True or False in the box.
  Always use minimal words.

Demonstration

1. Q: The context is: [paragraphs]. The question is: [question]
2. Let us think step by step.
3. A: `<think>`
   `Let's think through this step by step`
   `[shorthand reasoning]`
   `</think>`
   `\boxed{answer}`

Test example:

1. Q: The context is: [paragraphs]. The question is: [question]
2. Let us think step by step.
3. A: `<think>`
   `Let's think through this step by step`
   `[shorthand reasoning]`
   `</think>`
   `\boxed{answer}`

H.2 SoT Prompting with NWGM Approximation

For the prompt template, we design a unified structure applicable across all reasoning paradigms, reflecting our goal of building a general-purpose framework that supports diverse tasks without task-specific engineering. The task type is dynamically inferred by the pre-trained model. Unlike prior work (Zhang et al., 2025), which requires prompting the LLM to return task-specific answer formats, our template instead guides the model to perform symbolic reasoning and derive answers with minimal token usage through ICL.

Common Prompt Template

Instruction
You are a helpful assistant to perform [task type]. Based on the context, answer the question step by step and provide the final answer at the end. I will provide reasoning processes, and please improve them to ensure the correct answer.
Demonstration

1. Q: The question is: [question]
2. Let us think step by step,
3. The provided reasoning process is: [wrong_cot]
4. A: The improved reasoning process is: [correct_cot]
5. Therefore, the correct answer is: [answer]

Test example:

1. Q: The question is: [question]
2. Let us think step by step,
3. The provided reasoning process is: [r_k]
4. A: The improved reasoning process is: [improved_rs]
5. Therefore, the correct answer is: [answer]

I Case Study

This section presents two illustrative examples from CommonsenseQA and HotpotQA, highlighting the intermediate outputs at each stage of the framework. For each raw reasoning path, three improved reasoning paths are generated; however, for brevity, only the most informative ones are shown.

Case Study on CommonsenseQA

Question
Where do adults use glue sticks?
Choices {A: classroom, B: desk drawer, C: at school, D: office, E: kitchen drawer}
Cluster Reasoning Paths

- **Cluster 1** ($|C_0| = 3$, $P(r_0|do(X)) = 3/9 = 0.333$), $P(A = D \mid do(r_0)) = 0.333$

  Raw reasoning path: `<think> Let's think through this step by step #adults → office → desk </think> \boxed{office}`

  Improved reasoning path 1: `<improved_rs> #glue_stick → #adults → #office </improved_rs> \boxed{D}`

- **Cluster 2** ($|C_1| = 3$, $P(r_1|do(X)) = 3/9 = 0.333$), $P(A = D \mid do(r_1)) = 0.667$

  Raw reasoning path: `<think> Let's think through this step by step #glue_stick → children → school #adults → office </think> \boxed{office}`

  Improved reasoning path 1: `<improved_rs> #glue_stick → #adults → #office </improved_rs> \boxed{D}`

- **Cluster 3** ($|C_2| = 2$, $P(r_2|do(X)) = 2/9 = 0.222$), $P(A = C \mid do(r_2)) = 0.667$

  Raw reasoning path: `<think> Let's think through this step by step #glue_stick → common_use → school </think> \boxed{at school}`

  Improved reasoning path 1: `<improved_rs> #glue_stick`

→ #common_use → #school </improved_rs> \boxed{C}

- **Cluster 4** ($|C_3| = 1$, $P(r_3|do(X)) = 1/9 = 0.111$), $P(A = A \mid do(r_3)) = 1.000$

  Raw reasoning path: <think> Let's think through this step by step #adulthood → specific_tasks → work_supplies → desk_drawer → classroom_setting → glue_stick_use </think> \boxed{classroom}

  Improved reasoning path 1: <improved_rs> #glue_stick → #adult_tasks → #teaching → #classroom </improved_rs> \boxed{A}

Weighted Voting and Final Answer
Each cluster's answer is assigned a probability weight. The final answer is determined by summing these weights and selecting the answer with the highest total.

- $P(A = D|do(X)) = 0.333 \times 0.333 + 0.333 \times 0.667 = 0.3333$
- $P(A = C|do(X)) = 0.222 \times 0.667 = 0.1481$
- $P(A = A|do(X)) = 0.111 \times 1.000 = 0.1111$

The answer with the largest total weight is selected as the final answer.
Final Answer: \boxed{D}

Case Study on HotpotQA

Question
The driver know for doing backflips off his car lost to which driver in the 2009 NASCAR Nationwide Series?
Context With 25 top-five finishes, Kyle Busch was the season champion. He finished 210 points clear of Carl Edwards and 318 ahead of Brad Keselowski. Edwards is well known for doing a backflip off of his car to celebrate a victory, which was a result of saving himself from a potential fall when he had his first win.
Cluster Reasoning Paths

- **Cluster 1** ($|C_0| = 3$, $P(r_0|do(X)) = 3/8 = 0.375$), $P(A =$ **Kyle Busch** $\mid do(r_0)) = 3/3 = 1.000$

  Raw reasoning path: <think> Let's think through this step by step #Driver known for backflips off car → Edwards #Edwards → lost to Kyle Busch in 2009 NASCAR Nationwide Series </think> \boxed{Kyle Busch}

- **Cluster 2** ($|C_1| = 2$, $P(r_1|do(X)) = 2/8 = 0.250$), $P(A =$ **Carl Edwards** $\mid do(r_1)) = 2/3 = 0.667$

  Raw reasoning path: <think> Let's think through this step by step #Driver known for doing backflips → Carl Edwards #2009 NASCAR Nationwide Series winner → Kyle Busch #Closest competitor to the winner → Carl Edwards (210 points behind) #Driver who lost to the winner → Carl Edwards </think> \boxed{Carl Edwards}

  Improved reasoning path 2: [improved_rs] Let's think through this step by step: #Driver known for backflips → Carl Edwards (he is known for doing a backflip off of his car to celebrate a victory) #2009 NASCAR Nationwide Series winner → Carl Edwards (Edwards, not Kyle Busch, was the known driver for backflips) </improved_rs> \boxed{Carl Edwards}

  Improved reasoning path 3: Improved reasoning for the test example: #Driver known for doing backflips → Carl Edwards #Driver who lost to 2009 NASCAR Nationwide Series winner → Driver who lost to Kyle Busch Since the context mentions that Kyle Busch was the season champion with Carl Edwards and Brad Keselowski being the closest competitors, it can be inferred that Carl Edwards is the driver who lost to Kyle Busch. \boxed{Carl Edwards}

- **Cluster 3** ($|C_2| = 2$, $P(r_2|do(X)) = 2/8 = 0.250$), $P(A =$ **Kyle Busch** $\mid do(r_2)) = 2/3 = 0.667$

  Raw reasoning path: <think> Let's analyze the information: #Context mentions Carl Edwards is famous for backflips #Question asks who the backflip driver lost to in 2009 #Carl Edwards did not win the championship in 2009 #Therefore, Carl Edwards is the driver referenced as losing in the question </think> \boxed{Carl Edwards}

  Improved reasoning path 1: <improved_rs> Let's think through this step by step #Driver known for backflips off car → Edwards #Edwards lost to → Kyle Busch </improved_rs> \boxed{Kyle Busch}

  Improved reasoning path 3: <improved_rs> Let's think

through this step by step #Driver known for backflips off car → Edwards #Edwards' competition in 2009 NASCAR Nationwide Series → Kyle Busch </improved_rs> \boxed{Kyle Busch}

- **Cluster 4 ($|C_3| = 1$, $P(r_3|do(X)) = 1/8 = 0.125$), $P(A =$ Brad Keselowski $\mid do(r_3)) = 2/3 = 0.667$**

  Raw reasoning path: <think> Let's think through this step by step #Driver doing backflips → Carl Edwards #2009 NASCAR Nationwide Series → won by Kyle Busch #Driver who lost → not mentioned, but not Carl Edwards (who did backflips) </think> \boxed{not mentioned}

  Improved reasoning path 2: <improved_rs> #Driver doing backflips → Carl Edwards (from context) #2009 NASCAR Nationwide Series → won by Kyle Busch #Kyle Busch finished 318 points ahead of Brad Keselowski #Driver who lost to the champion → Brad Keselowski </improved_rs> \boxed{Brad Keselowski}

  Improved reasoning path 3: [improved_rs] The improved reasoning path is: #Driver doing backflips → Carl Edwards #2009 NASCAR Nationwide Series → won by Kyle Busch #Driver ranked 3rd → Brad Keselowski (the driver one step behind Carl Edwards, who is one step behind Kyle Busch) The correct answer is therefore: \boxed{Brad Keselowski}

Weighted Voting and Final Answer
Each cluster's answer is assigned a probability weight. The final answer is determined by summing these weights and selecting the answer with the highest total.

- $P(A = \text{Kyle Busch}|do(X)) = 0.375 \times 1.000 + 0.250 \times 0.667 = 0.5417$
- $P(A = \text{Carl Edwards}|do(X)) = 0.250 \times 0.667 = 0.1667$
- $P(A = \text{Brad Keselowski}|do(X)) = 0.125 \times 0.667 = 0.0833$

The answer with the largest total weight is selected as the final answer.
Final Answer: \boxed{Kyle Busch}