Causal Prompting: Debiasing Large Language Model Prompting based on Front-Door Adjustment

Congzhi Zhang[♠♢] **Linhai Zhang**[♠♢] **Jialong Wu**[♠♢] **Deyu Zhou**[♠♢*] **Yulan He**[♡♣]
[♠] School of Computer Science and Engineering, Southeast University, Nanjing, China
[♢] Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
[♡] Department of Informatics, King's College London, UK
[♣] The Alan Turing Institute, UK
{zhangcongzhi, lzhang472, jialongwu, d.zhou}@seu.edu.cn
yulan.he@kcl.ac.uk

Abstract

Despite the notable advancements of existing prompting methods, such as In-Context Learning and Chain-of-Thought for Large Language Models (LLMs), they still face challenges related to various biases. Traditional debiasing methods primarily focus on the model training stage, including approaches based on data augmentation and reweighting, yet they struggle with the complex biases inherent in LLMs. To address such limitations, the causal relationship behind the prompting methods is uncovered using a structural causal model, and a novel causal prompting method based on front-door adjustment is proposed to effectively mitigate LLMs biases. In specific, causal intervention is achieved by designing the prompts without accessing the parameters and logits of LLMs. The chain-of-thought generated by LLM is employed as the mediator variable and the causal effect between input prompts and output answers is calculated through front-door adjustment to mitigate model biases. Moreover, to accurately represent the chain-of-thoughts and estimate the causal effects, contrastive learning is used to fine-tune the encoder of chain-of-thought by aligning its space with that of the LLM. Experimental results show that the proposed causal prompting approach achieves excellent performance across seven natural language processing datasets on both open-source and closed-source LLMs.

1 Introduction

Large Language Models (LLMs) have shown remarkable emergent abilities, including In-Context Learning (ICL) [Brown et al., 2020] and Chain-of-Thought (CoT) prompting [Wei et al., 2022, Wang et al., 2022], which allow LLMs to perform natural language tasks based on only a few instances without weight updating. These prompting methods have achieved significant results across many traditional natural language processing tasks, including sentiment analysis, natural language inference, and machine reading comprehension [Kojima et al., 2022, Zhou et al., 2022, Liu et al., 2023].

However, recent studies have shown that these advanced prompting methods are not robust enough [Ye et al., 2023] and can lead LLMs to produce hallucinatory results with incorrect or unfaithful intermediate reasoning steps [Lyu et al., 2023, Wang et al., 2023a, Bao et al., 2024, Turpin et al., 2024].

[*] Corresponding Author.

Preprint. Under review.

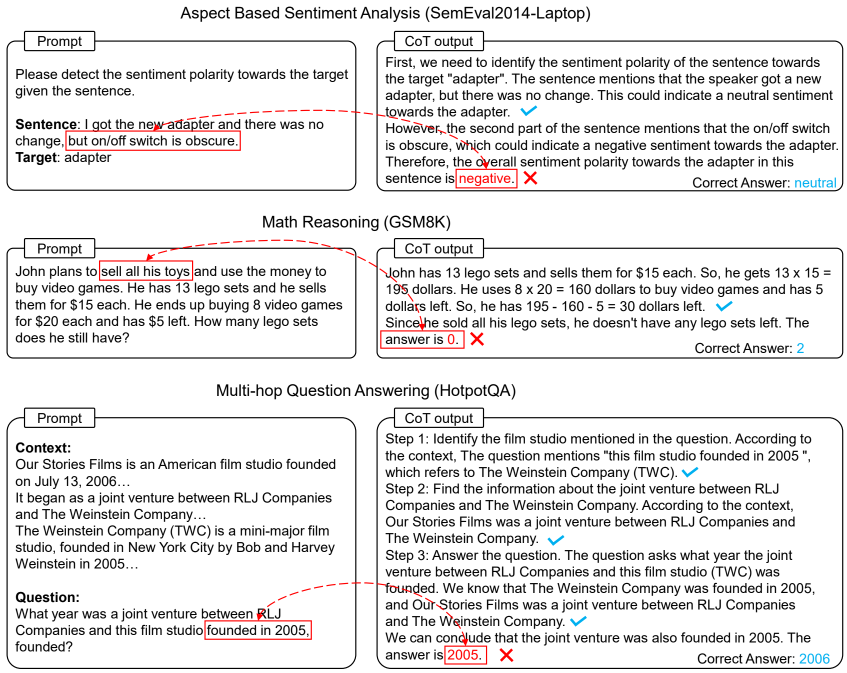


Figure 1: LLMs suffer from bias in the pertaining corpus, leading them to rely on irrelevant text spans in prompts and generating incoherent chain-of-thoughts that harm the logical reasoning capability of the model. These examples were obtained by using the CoT prompting [Wei et al., 2022] on the LLaMA3-8B model.

Some studies [Mallen et al., 2023, Wang et al., 2023b] believe that this phenomenon is due to the a conflict between the internal knowledge bias of LLMs and the external knowledge. Therefore, an effective solution is to interact with an external knowledge base to validate and adjust the reasoning process of LLMs [Wang et al., 2023a, Zhang et al., 2023a]. Moreover, recent work debiases the chain-of-thoughts of LLMs by incorporating counterfactual knowledge and causal interventions [Lanham et al.]. However, these methods are specifically tailored for knowledge-intensive tasks. Bias problems are also observed in other NLP tasks. As shown in Figure 1, in aspects-based sentiment analysis, mathematical reasoning, and multi-hop question-answering tasks, LLMs sometimes overly depend on certain text spans in the prompts, leading to wrong reasoning and answers. Notably, the first two tasks mentioned are not knowledge-intensive. We argue that LLMs fail to capture the true causal effect between questions and reasoning results and instead establish spurious correlations between certain text spans and answers.

In addition to the above qualitative analysis, our quantitative experiments also show that the current prompting methods are ineffective in addressing the bias issue. As shown in Figure 2, the performance of all prompting methods drops significantly when evaluated on the corresponding adversarial dataset compared to the original dataset, indicating that LLMs may suffer from bias in the pertaining corpus. Moreover, it has been demonstrated that LLMs exhibit label bias, recency bias, and entity bias from context [Zhao et al., 2021, Wang et al., 2023c, Fei et al., 2023].

Traditional debiasing methods mitigate the bias issue mainly during the model training stage, utilizing approaches such as data augmentation-based [Wei and Zou, 2019, Lee et al., 2021] and reweighting [Schuster et al., 2019, Mahabadi et al., 2019]. Data augmentation-based methods face challenges due to the cost and complexity of annotating bias cases, particularly limited by context length. Reweight-based methods encounter difficulties in assigning weights to each sample in prompt-based learning scenarios. Recently, debias methods based on causal inference [Pearl et al., 2000, Pearl, 2022] have become popular because of their strict theoretical guarantees and good generalization. Causal inference-based methods only need to calibrate model prediction results during the inference stage [Niu et al., 2021, Tian et al., 2022, Guo et al., 2022, Xu et al., 2023, Chen et al., 2023a], which makes them well-suited for prompt-based learning scenarios. However, counterfactual inference requires accessing LLM output logits, while back-door adjustment requires specific confounding variable values.

To address the aforementioned challenge, we propose to debias prompting methods through causal intervention using front-door adjustment [Pearl et al., 2016]. Front-door adjustment enables causal intervention without the need to access confounding variable values or LLM output logits. As shown in Figure 3(a), the causal relationship behind the prompting method is uncovered using a structural causal model. Here X denotes the input prompt, comprising demonstrations and test examples. A denotes the predicted answer generated by the LLM. U is the unobservable confounder that introduces various biases in the pertaining corpus.

The debiasing process involves measuring the causal effect between the treatment X and the outcome A. However, as U absorbs complex biases of LLMs that are difficult to model or detect, back-door adjustment is not feasible for calculating the causal effect between X and A. To address this issue, as shown in Figure 3(b), we use the chain-of-thought generated by LLM as the mediator variable R between X and A.

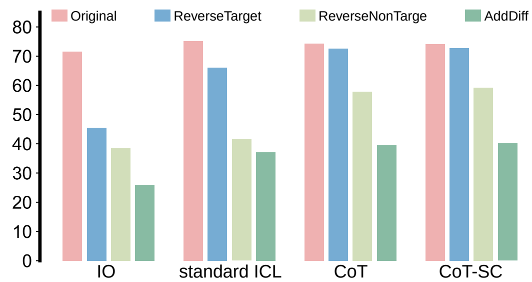


Figure 2: Performance of different prompting methods on ABSA [Pontiki et al., 2016] and its adversarial datasets on LLaMA-7b. ReverseTarget, ReverseNonTarget, and AddDiff denote three different adversarial transformations by TextFlint [Wang et al., 2021]. IO denotes the zero-shot setting where only the input question outputs the answer.

As Figure 1 illustrates, while LLMs initially reason correctly, biases often confuse the final step of answer derivation. To simplify, we ignore the edges between U and R, aligning our causal graph with the front-door criterion [Pearl et al., 2016]. By this way, we can use the front-door adjustment to estimate the causal effect between X and A without accessing U.

Therefore, in this paper, we propose **Causal Prompting**, a novel prompting method for debiasing based on front-door adjustment. Unlike previous causal inference-based methods, causal intervention is implemented by modifying prompts without accessing the parameters and logits of LLMs. Specifically, to estimate the causal effect between X and R, we leverage self-consistency (SC) [Wang et al., 2022] of LLMs and a clustering algorithm to compute the probability of the chain-of-thought R. To measure the causal effect between R and A, we use the normalized weighted geometric mean (NWGM) approximation [Xu et al., 2015] to select the optimal demonstration set, which can help the model to generate an unbiased answer. Overall, CoT, SC, and ICL are effectively combined through front-door adjustment to mitigate LLM biases in NLP tasks. Note that in the clustering and NWGM algorithms, an $\mathrm{Encoder}$ is needed to obtain the representations of chain-of-thoughts. Since $\mathrm{Encoder}$ and LLMs have different semantic understanding of the chain-of-thought, we use contrastive learning [Chen et al., 2020] to fine-tune the $\mathrm{Encoder}$ to align its representation space with LLMs to estimate causal effects more accurately.

The contributions of this work are summarized as follows:

- Our work aims to identify and analyze the bias problem in LLM prompting methods from the perspective of causal inference, adhering more closely to the principles of the field. Moreover, the front-door adjustment is proposed to theoretically address the bias problem in prompting.
- Contrastive learning is proposed to fine-tune the $\mathrm{Encoder}$ of the chain-of-thoughts, aligning the space of the $\mathrm{Encoder}$ with LLMs to accurately capture representations of chain-of-thoughts and estimate causal effects.
- The proposed approach achieves excellent performance across seven natural language processing datasets using both open-source and closed-source LLMs.

2 Preliminaries

2.1 Structural Causal Model and Causal Intervention

A Structural Causal Model (SCM) [Pearl et al., 2016] is used to describe the causal relationships between variables. In SCM, we typically use a directed acyclic graph $G = \{V, E\}$, where V represents the set of variables and E represents the set of direct causal relationships.

As shown in Figure 3(a), X denotes the input prompt, including demonstrations and test examples. A denotes the predicted answer generated by the LLMs. LLMs generate answers based on prompt, so we have $X \to A$, which means that X is the direct cause of A. LLMs might learn spurious correlations between text patterns and answers from pre-trained corpora or instruction-supervised fine-tuning datasets [Xing et al., 2020, Li et al., 2024, Bao et al., 2024], leading to bias in downstream tasks. Previous work argues that the reason for this bias is that LLMs tend to follow a certain latent concept [Xie et al., 2021] or an implicit reasoning results [Li et al., 2024] in the reasoning process, rather than following the explicitly generated chain-of-thought. This leads to the final answer does not necessarily follow from the generated chain-of-thought, specifically, there is no actual causal relationship between the chain-of-thought and the answer [Lyu et al., 2023, Bao et al., 2024]. To accurately calculate the causal effect between X and A, we use the unobservable variable U to describe this latent concept or implicit reasoning results, using the back-door path $X \leftarrow U \to A$ denotes that the causality of X and A is confounded by U.

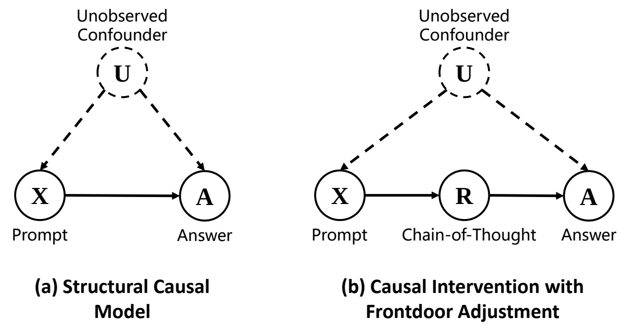


Figure 3: **Structural causal model for the prompting method.** (a) The causality of prompt and answer is confounded by unobservable variable. (b) The chain-of-thought generated by LLMs as a mediator variable between prompt and answer.

In SCM, if we want to compute the true causal effect between two variables X and A, we should block every back-door path between them [Pearl and Mackenzie, 2018]. For example, as shown in Figure 3(a), we should block $X \leftarrow U \to A$ to obtain the true causal effect between X and A. We typically use causal interventions for this purpose, which use the do operation to estimate the causal effect between X and A. In the causal graph satisfying Figure 3(a), the do-operation can be computed by back-door adjustment [Pearl et al., 2016]:

$$P(A|do(X)) = \sum_u P(A|X,u)P(u) \tag{1}$$

2.2 Front-door Adjustment

Since confounding factor U is inaccessible, back-door adjustment cannot be performed. Fortunately, the front-door adjustment [Pearl et al., 2016] does not require access to the values of the confounding factor U to calculate the causal effect between X and A. As shown in Figure 3(b), we use the chain-of-thought generated by LLM as a mediator variable R between X and A.

In practice, as depicted in Figure 1, LLM can perform correct reasoning at the beginning, but it is often easily confused by bias in the last step of deriving the answer. Consequently, we decided to start with the simple SCM and focus on the confounder between X and A. In order to simplify the causal graph, we ignore the confounder of R with other variables, aligning our causal graph with the front-door criterion [Pearl et al., 2016]. According to the front door adjustment, $P(A|do(X))$ can be formulated as:

$$P(A|do(X)) = \sum_r P(A|do(r))P(r|do(X)) \tag{2}$$

where $r \in R$ is the chain-of-thought generated by LLMs in response to the prompt X. The causal effect between X and A is decomposed into two partially causal effects $P(r|do(X))$ and $P(A|do(r))$.

Next, we discuss how to estimate these two components separately. The first component is $P(r|do(X))$, represents the probability distribution of the chain-of-thought r given the intervention $do(X)$. To compute $P(r|do(X))$, we need to block the backdoor path $X \leftarrow U \to A \leftarrow R$ between X and R. Since there exists a collision structure $U \to A \leftarrow R$, the backdoor path has been blocked [Pearl et al., 2016] and we can get:

$$P(r|do(X)) = P(r|X) \tag{3}$$

Now, we focus on the computation of the second component $P(A|do(r))$, represents the probability distribution of the answer A given the intervention $do(r)$. To compute $P(A|do(r))$, we need to block the backdoor path $R \leftarrow X \leftarrow U \rightarrow A$ between R and A. Since we do not have access to the details of U, we implement back-door adjustments with the help of prompt X:

$$P(A|do(r)) = \sum_{x} P(x)P(A|r, x) \tag{4}$$

where $x \in X$ denotes the input prompt, including demonstrations and test examples.

Finally, substituting Equations (3) and (4) into Equation (2) after we obtain the estimation of $P(r|do(X))$ and $P(A|do(r))$. Hence, the final $P(A|do(X))$ can be represented as follows:

$$P(A|do(X)) = \sum_{r} P(r|do(X))P(A|do(r)) = \underbrace{\sum_{r} P(r|X)}_{CoT-SC} \underbrace{\sum_{x} P(x)P(A|r, x)}_{ICL} \tag{5}$$

where the first component $\sum_r P(r|do(X))$ can be estimated by combining the CoT and SC prompting methods, and the second component $P(A|do(r))$ can be computed by selecting the demonstration examples in ICL prompting.

3 Method

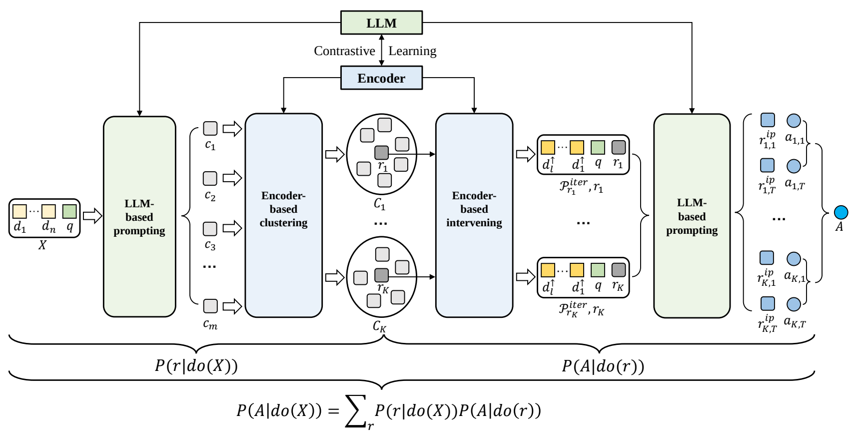


Figure 4: **The overall framework of Causal Prompting**. Firstly, based on the input prompt X consisting of the demonstration examples and a question of the test example, we query the LLM to generate m distinct CoTs. Then, these CoTs are clustered into K clusters by an Encoder-based clustering algorithm. Subsequently, K representative CoTs are selected by searching the closest CoT to the cluster center. Secondly, the optimal demonstration examples are retrieved for each representative CoT through the Encoder-based intervention algorithm, and then the input prompt $\mathcal{P}_{r_k}^{iter}$ after the intervention is obtained. Finally, we query the LLM T times, obtaining T improved CoTs and T answers for each representative CoT. The final answer is obtained by performing a weighted voting.

As shown in Figure 4, Causal Prompting aims to estimate the causal effect between input X and answer A. The estimation is achieved using the front-door adjustment, which divides the causal pathway into **two** distinct parts: the causal effect between X and chain-of-thought r, and the causal effect between r and A.

First, the causal effect between X and chain-of-thought r, $P(r|do(X))$ is estimated by combining the Chain-of-Thought prompting with a Encoder-based clustering algorithm. **Second**, the causal effect between r and A, $P(A|do(r))$ is estimated by combining the In-Context Learning prompting with the normalized weighted geometric mean (NWGM) approximation algorithm. The final answer is aggregated by performing a weighted voting algorithm. Moreover, contrastive learning[Chen et al., 2020, Gao et al., 2022, Zhang et al., 2023b] is employed to align the representation space of the Encoder and the LLMs for more precise estimation.

We will first introduce the estimation of $P(r|do(X))$ and $P(A|do(r))$, respectively, then combine them to derive $P(A|do(X))$. Finally, we will discuss how we align the representation space between the Encoder and the LLM.

3.1 Estimation of $P(r|do(X))$

We firstly undertake the estimation of $P(r|do(X))$. $P(r|do(X))$ measures the causal effect between input X and chain-of-thought r. As shown in Eq. (3), the estimation of $P(r|do(X))$ is equivalent to the estimation of $P(r|X)$. However, $P(r|X)$ is still intractable for LLMs. On the one hand, the output probability is often inaccessible for most closed-source LLMs; on the other hand, the chain-of-thoughts r are challenging to enumerate comprehensively. Therefore, to estimate the causal effect $P(r|do(X))$ for both open-source and closed-source LLMs, we employ the CoT prompting and integrate it with a clustering algorithm. To be more specific, we initially prompt the LLMs to generate multiple CoTs based on the input. The prompts for CoTs generation are detailed in Appendix H.1. Subsequently, the CoTs are projected into embeddings. The embeddings are then clustered to form distinct groups based on their similarity. Finally, the centroid of each cluster is selected as the optimal and representative chain-of-thought. The probability associated with each representative chain-of-thought is then estimated based on the size of its respective cluster.

To enhance the quality of generated CoTs, n in-context demonstrations d are selected from training set based on question similarity. These demonstrations are then concatenated with the test question q^{test} to form the final prompt. Thus, the final prompt $\mathcal{P}$ is structured as follows:

$$\mathcal{P} = [d_1, ..., d_n, q^{test}] \tag{6}$$

where each $d_i = (q_i^{demo}, r_i^{demo})$ contain the demonstration question q_i^{demo} and its corresponding demonstration chain-of-thought r_i^{demo}. Where $i \in \{1, ..., n\}$, n denotes the number of demonstration examples in few-shot prompt method. In the practical implementation, we use prompt $\mathcal{P}$, which is fed into the LLMs to represent X in the structural causal model.

Based on the input prompt $\mathcal{P}$, LLMs are prompted to generate m distinct CoTs c by increasing the temperature parameter of LLMs. This adjustment encourages more diverse outputs, where the same procedure is also employed in self-consistency prompting of LLMs Wang et al. [2023d]. In this way, we can obtain the set of chain-of-thoughts as follows:

$$\{c_i | i = 1, ..., m\} = \mathrm{LLM}(\mathcal{P}) \tag{7}$$

To perform the distance-based clustering method, the generated CoT c_i are further fed into a $\mathrm{Encoder}$ to get the text embedding $\overline{c}_i$. Following the previous work [Devlin et al., 2018], the input is concatenated with the special tokens [CLS] and [SEP], and the embedding of the [CLS] token is taken as the embedding of CoT c_i.

$$\overline{c}_i = \mathrm{Encoder}([\mathrm{CLS}], c_i, [\mathrm{SEP}]) \tag{8}$$

Then K-means clustering algorithm [Har-Peled and Kushal, 2005, Wu et al., 2023] is applied to the embeddings to get k clusters C as follows:

$$\{C_1, ..., C_K\} = \text{K-means}(\overline{c}_1, ..., \overline{c}_m) \tag{9}$$

where C_k refers to the k-th cluster of the clustering result, K denotes the number of clusters.

Based on the clusters, K representative chain-of-thoughts r are selected by searching the closest chain-of-thought to the cluster center.

$$r_k = \mathrm{Center}(C_k), k = 1, ..., K \tag{10}$$

The causal effect between input X and chain-of-thought r_k is estimated based on the cluster size as follows:

$$P(r_k|do(X)) \approx \frac{|C_k|}{m} \tag{11}$$

where $|C_k|$ denotes the size of cluster C_k.

3.2 Estimation of $P(A|do(r))$

Based on the K chain-of-thoughts selected by Equation (10) in Section 3.1, we estimate $P(A|do(r_k))$ for each chain-of-thought r_k. For convenience, we omit the subscript k and use $P(A|do(r))$ to denote $P(A|do(r_k))$ in the following. $P(A|do(r))$ measures the causal effect between the chain-of-thought r and the answer A. Based on the discussion in Equation (4), $P(A|do(r))$ can be calculated with backdoor adjustment as follows:

$$P(A|do(r)) = \sum_{x \in X} P(x)P(A|r, x) = \mathbb{E}_{x \in X}[P(A|r, x)] \tag{12}$$

where $P(A|r, x)$ denotes the probability of the final answer A generated by LLM based on the given prompt x and the chain-of-thought r.

However, the value space of X is inexhaustible in most of the cases, and previous work employs the normalized weighted geometric mean (NWGM) approximation [Xu et al., 2015, Tian et al., 2022, Chen et al., 2023a] to tackle this problem, where a confounder embedding $\overline{x}^{'}$ is estimated to approximate the expectation of variable X.

$$\mathbb{E}_{x\in X}[P(A|r,x)] \approx P(A|r,\mathbb{E}_{x\in X}[x]) \approx P(A|concat(r,\overline{x}^{'})) \tag{13}$$

where $concat(\cdot,\cdot)$ denotes vector concatenation, $\overline{x}^{'}$ denotes the confounder embedding of X.

Inspired by the previous works [Xu et al., 2015, Tian et al., 2022, Chen et al., 2023a, Zhang et al., 2024], we propose a prompt version of NWGM approximation to perform the back-door adjustment for LLMs prompting by combining a Encoder-based intervention and In-Context Learning (ICL) prompting. The original idea of NWGM is to augment the representation of the chain-of-thought r with an embedding $\overline{x}^{'}$ that contains all sample information as much as possible. However, at the prompt level, we cannot include all samples in context due to the limited context length, so we use only those samples that are most useful for improving the current chain-of-thought r.

Specifically, we use the Encoder to obtain the embedding $\overline{r}_k$ of the k-th chain-of-thought r_k. Subsequently, ICL demonstrations are selected by searching the entire training set based on the chain-of-thought embedding $\overline{r}_k$ to approximate the effect of taking expectations on input X. Finally, we rank the ICL demonstrations according to their similarity weights to indicate the importance of different samples.

Note that, as shown in Equation (6), the input prompt $\mathcal{P}$ includes demonstrations d and test question q^{test}. Directly modifying the certain text span of test examples will change the semantics of question q^{test}. Therefore, we only modify the demonstrations d and implement the NWGM approximation by In-Context Learning. In fact, the goal of our prompt version of the NWGM algorithm is to enable the LLMs to learn from the demonstrations how to improve the chain-of-thought r of the test example. As shown in the prompt template in Appendix H.2, we introduce both wrong and correct chain-of-thoughts of demonstrations.

Given a training set $\mathcal{D} = \{d_j = (q_j, r_j^{wrong}, r_j^{correct})\}_{j=1}^{N}$, and a chain-of-thought r_k of test example, where q_j denotes the question of j-th training sample, r_j^{wrong} and $r_j^{correct}$ denote the wrong and correct chain-of-thoughts of demonstration d_j, r_k refers to the k-th chain-of-thought selected by Equation (10) in Section 3.1, N denotes the size of the training set. The embedding $\overline{r}_k$ of chain-of-thought r_k and the embedding $\overline{d}_j$ of demonstration d_j are obtained by the following:

$$\begin{aligned}\overline{r}_k &= \text{Encoder}(\text{[SEP]}, r_k, \text{[SEP]})\\ \overline{d}_j &= \text{Encoder}(\text{[CLS]}, r_j^{wrong}, \text{[SEP]})\end{aligned} \tag{14}$$

Previous works [Margatina et al., 2023, Liu et al., 2022] have shown that using demonstration examples that are semantically similar to the test examples allows better performance for In-Context Learning. Therefore, the back-door intervention is approximated by searching the most similar instance based on chain-of-thought embedding $\overline{r}_k$. Specifically, we sort the training set $\mathcal{D}$ from largest to smallest according to the cosine similarity between $\overline{r}_k$and $\overline{d}_j$.

$$\{d_j^{\uparrow}\}_{j=1}^{N} = Sort(\mathcal{D}, \overline{r}_k, \{\overline{d}_j\}_{j=1}^{N}) \tag{15}$$

where $d_j^{\uparrow}$ denotes the sorted demonstration example, $Sort$ means that, given a predefined cosine similarity function cos, the samples are ordered so that $cos(\overline{r}_k, \overline{d}_i) \geq cos(\overline{r}_k, \overline{d}_j)$ when $i < j$.

Then the l most similar demonstration examples are selected to concatenate into prompt, where $l \ll N$. Note that, unlike the KATE [Liu et al., 2021] method, we put the most similar demonstration samples closer to the test samples because this order is more beneficial for our NWGM algorithm to learn information for improving the chain-of-thoughts from the demonstration based on practical experiments, detailed in Appendix E.2. For each chain-of-thought r_k of a test sample, the final input prompt after intervention is given as follows:

$$\mathcal{P}_{r_k}^{iter} = [d_l^{\uparrow}, ..., d_1^{\uparrow}, q^{test}] \tag{16}$$

Subsequently, we query the LLMs T times, obtaining T answers and T improved chain-of-thoughts using the prompt $\mathcal{P}_{r_k}^{iter}$ and chain-of-thought r_k.

$$\{(r_{k,t}^{ip}, a_{k,t}) | t = 1, ..., T\} = \text{LLM}(\mathcal{P}_{r_k}^{iter}, r_k) \tag{17}$$

where $r_{k,t}^{ip}$ denotes the t-th improved chain-of-thought for chain-of-thought r_k.

We then use majority voting to estimate the probability of the answer as follows:

$$P(A|do(r_k)) \approx \frac{\sum_{t=1}^{T} \mathbb{I}(A = a_{k,t})}{T} \tag{18}$$

3.3 Estimation of $P(A|do(X))$

Based on the results of Equation (11) in Section 3.1 and Equation (18) in Section 3.2, the final answer is obtained by performing a weighted voting as follows:

$$P(A|do(X)) = \sum_{r_k} P(r_k|do(X))P(A|do(r_k)) = \sum_{k=1}^{K} \frac{|C_k|}{m} \cdot \frac{\sum_{t=1}^{T} \mathbb{I}(A = a_{k,t})}{T} \tag{19}$$

Finally, we chose the answer with the largest weight as the final answer. In this way, with the front-door adjustment, we calibrate the probability distribution $P(A|X)$ obtained by the CoT-SC method to $P(A|do(X))$ obtained by the Causal Prompting method. Algorithm 1 in Appendix B shows the overall prompting process. Cases in Appendix G show the overall flow and intermediate step output of Causal Prompting on mathematical reasoning and multi-hop question answering datasets.

3.4 Representation Space Alignment

In the clustering discussed in Section 3.1 and NWGM algorithm presented in Section 3.2, an $\mathrm{Encoder}$ is needed to derive the representations of chain-of-thoughts. However, the semantic representation of $\mathrm{Encoder}$ and LLM differ significantly. Two chain-of-thoughts that LLM considers similar may not be close in the representation space of the $\mathrm{Encoder}$. As illustrated in Figure 7 in Appendix A, the chain-of-thoughts generated by LLM are not distinctly separable in the representation space of the vanilla $\mathrm{Encoder}$.

To align the representation spaces of the $\mathrm{Encoder}$ and the LLMs, we take each chain-of-thought r in the training dataset $\mathcal{D}$ as an anchor, use LLM to generate the corresponding positive samples, use the other samples within the batch as negative samples, and then use contrastive learning to fine-tune the $\mathrm{Encoder}$. The prompt template used to generate positive samples is detailed in Appendix H.3.

For chain-of-thought r, we prompt the LLM to generate a similar sentence r^+ as the positive sample. Following previous works [Gao et al., 2022, Zhang et al., 2023b], we use the InfoNCE loss [Chen et al., 2020] to fine-tune the $\mathrm{Encoder}$:

$$\sum_{\overline{r}_p \in Pos(r)} -log \frac{g(\overline{r}, \overline{r}_p)}{g(\overline{r}, \overline{r}_p) + \sum_{j \in Neg(r)} g(\overline{r}, \overline{r}_j)} \tag{20}$$

where the $\overline{r}$ and $\overline{r}_p$ are the representations of r and its positive samples. $Pos(r)$ and $Neg(r)$ refer to the positive set and the negative set for the chain-of-thought r. $Pos(r) = \{\overline{r}_{p1}, \overline{r}_{p2}\}$, where $\overline{r}_{p1}$ is augmented representation of the same chain-of-thought r, obtained with different dropout masks, and $\overline{r}_{p2}$ is the representation of positive sample r^+. $j \in Neg(r)$ is the index of in-batch negative samples. g is a function: $g(\overline{r}, \overline{r}_p) = exp(\overline{r}^T \overline{r}_p / temp)$, where $temp$ is a positive value of temperature in the contrastive learning.

4 Experiments

4.1 Datasets

We evaluate the effectiveness of our approach on three tasks: **Math Reasoning** (GSM8K [Cobbe et al., 2021], MATH [Hendrycks et al., 2021]), **Multi-hop Question Answering** (HotpotQA [Yang et al., 2018], MuSiQue [Trivedi et al., 2022]), and **Natural Language Understanding** (Aspect-based Sentiment Analysis (ABSA) [Pontiki et al., 2016], Natural Language Inference (NLI) [Williams et al., 2017], and Fact Verification (FV) [Thorne et al., 2018]). For the NLU tasks, we use the original datasets (in-distribution, ID) and the corresponding adversarial datasets (out-of-distribution, OOD) to verify the robustness of our method. Further details regarding the datasets are provided in Appendix C.4. The details regarding the evaluation can be found in Appendix C.5.

4.2 Baselines

We compare our approach with three other few-shot prompting approaches to evaluate its effectiveness: Standard ICL, CoT and CoT-SC. Their detailed settings are presented in Appendix C.1. Detailed settings and implementations of our method **Causal Prompting** can be found in Appendix C.2 and Appendix C.3.

4.3 Main Results

Table 1 shows the comparison results between causal prompting and the aforementioned baselines. Expectedly, the performance of Standard ICL, CoT, and CoT-SC improves progressively, as each subsequent method is an enhanced version of its predecessor. It not only confirms the effectiveness of integrating CoT into ICL, consistent with [Brown et al., 2020, Wei et al., 2022, Zhou et al., 2022], but also validates the efficacy of employing multiple

Table 1: The comparison results of **Causal Prompting** against baselines across different backbone LLMs, including LLaMA2, LLaMA3 and GPT-3.5, on seven datasets. The best results are in **bold**.

	GSM8K	MATH	HotpotQA		MuSiQue		ABSA	NLI	FV
Method	Acc	Acc	EM	F1	EM	F1	Acc	Acc	Acc
LLaMA2									
Standard ICL	6.14	3.71	41.20	59.56	26.09	41.16	47.26	28.20	56.87
CoT	27.07	4.72	44.70	64.84	18.71	30.27	49.12	27.56	70.07
CoT-SC	31.92	6.32	49.30	68.53	31.16	46.36	53.70	33.57	72.20
Causal Prompting	**36.47**	**8.76**	**52.20**	**70.88**	**34.68**	**48.79**	**67.55**	**50.83**	**81.07**
LLaMA3									
Standard ICL	18.65	14.24	37.20	62.17	17.42	24.22	72.14	63.75	80.67
CoT	74.07	40.35	48.90	72.75	38.88	54.38	71.55	64.19	81.80
CoT-SC	82.41	56.61	52.70	75.43	41.37	59.78	75.92	65.15	83.87
Causal Prompting	**87.95**	**62.76**	**58.50**	**78.18**	**48.07**	**64.23**	**79.06**	**67.97**	**86.67**
GPT-3.5									
Standard ICL	33.74	23.08	2.10	3.68	28.84	39.27	69.26	53.52	75.33
CoT	71.87	53.50	11.70	16.49	41.37	57.82	65.74	63.55	80.67
CoT-SC	80.21	58.38	41.60	56.82	46.27	60.83	74.59	66.88	82.73
Causal Prompting	**85.44**	**70.18**	**58.20**	**78.10**	**50.13**	**65.40**	**80.13**	**71.93**	**86.53**

Table 2: The results of the robustness study on LLaMA3. `Ori` denotes the original dataset (ID) and `Adv` denotes the adversarial dataset (OOD). The best results are in **bold**.

	ABSA		NLI		FV	
Methods	`Ori`	`Adv`	`Ori`	`Adv`	`Ori`	`Adv`
Standard ICL	75.71	70.30	76.30	50.27	90.00	76.00
CoT	77.27	68.60	74.81	54.77	91.40	77.00
CoT-SC	**80.56**	73.53	76.17	57.16	93.40	79.10
Causal Prompting	79.78	**78.69**	**76.67**	**58.62**	**95.40**	**82.30**

sampling and voting strategies [Wang et al., 2022]. **Causal Prompting** consistently delivers the best results across all metrics and datasets. It indicates that our prompting method can comprehensively improve the ability of LLM in all three tasks. Specifically, our method exhibits a more pronounced improvement in Math Reasoning and Multi-hop Question Answering tasks, with an average performance enhancement of approximately **5%-10%**. This substantial increase underscores our method's greater efficacy in tackling more challenging problems.

4.4 Robustness Study

To verify the robustness of our method, we further test the performance of **Causal Prompting** on the original data and adversarial data of the NLU task, respectively. Tables 2 show the performance comparison results of our method and baselines on LLaMA3 model. Although the performance of Causal Prompting decreases on `Ori` of ABSA, the improvement is larger on `Adv` data, resulting in the highest overall performance, see in 1. This phenomenon aligns with findings reported in previous work on causal inference [Tian et al., 2022, Wang et al., 2023c]. It can be observed that the `Adv` of Causal Prompting is the highest on all datasets. This shows that our method generalizes well for both synthetic adversarial data in ABSA and NLI generated by TextFlint [Wang et al., 2021] and human-annotated real adversarial data in FV. This further validates the robustness of our model in handling datasets with significant bias. The robustness studies of LLaMA2 and GPT3.5 are presented in Appendix E.1.

5 Conclusion

We introduced Causal Prompting, a novel method for debiasing LLMs in NLP tasks by utilizing front-door adjustment in this work. The CoT generated by LLMs is employed as a mediator variable in the causal graph. Specifically, the causal effect between input prompt and output answer is decomposed into two distinct components, the causal effect from the input prompt to CoTs and from CoTs to the answer. The former component is estimated by combining the CoT prompting with a Encoder-based clustering algorithm. The latter component is estimated by combining the ICL prompting with the NWGM approximation algorithm. Moreover, Contrastive learning is used to fine-tune the Encoder so that the representation space of the Encoder is aligned with the LLM to estimate the causal effect more accurately. Our experimental results demonstrate that Causal Prompting significantly improves performance across seven NLP tasks on both open-source and closed-source LLMs.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

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*, 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 Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *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, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.20. URL `https://aclanthology.org/2023.ijcnlp-main.20`.

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*, 2023a.

Guangsheng Bao, Hongbo Zhang, Linyi Yang, Cunxiang Wang, and Yue Zhang. Llms with chain-of-thought are non-causal reasoners. *arXiv preprint arXiv:2402.16048*, 2024.

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, 2024.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL `https://aclanthology.org/2023.acl-long.546`.

Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. Resolving knowledge conflicts in large language models. *arXiv preprint arXiv:2310.00935*, 2023b.

Shuo Zhang, Liangming Pan, Junzhou Zhao, and William Yang Wang. Mitigating language model hallucination with interactive question-knowledge alignment. *arXiv preprint arXiv:2305.13669*, 2023a.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson E. Denison, Danny Hernan-720 dez, Dustin Li, Esin Durmus, Evan Hubinger, Xingxuan Li, Yew Ruochen Zhao, Bosheng Ken Chia, Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Meng Wang, Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, Alex Troy Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Stephen L. Morgan, Christopher Winship, Weijia Shi, Xiaochuang Han, Mike Lewis, Luke Tsvetkov, Zettlemoyer Scott, Wen-tau, Xin Su, Tiep Le, Steven Bethard, Yifan Kai Sun, Ethan Xu, Hanwen Zha, Yue Liu, Hugo Touvron, Louis Martin, Kevin Stone, Peter Al-845 bert, Amjad Almahairi, Yasmine Babaei, Nikolay, Cunxiang Wang, Xiaoze Liu, Xian-871 Yuanhao Yue, Keheng Wang, Feiyu Duan, Peiguang Sirui Wang, Junda Wu, Tong Yu, Shuai Li, Deconfounded, Suhang Wu, Min Peng, Yue Chen, Jinsong Su, Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Ben-959, William Cohen, Ruslan Salakhutdinov, Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Chenhan Yuan, Qianqian Xie, Jimin Huang, Li-Yan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng, Fangyuan Gao, Xingyi Zou, Heng Cheng, and Ji. Decot: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. URL `https://api.semanticscholar.org/CorpusID:268290124`.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR, 2021.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. A causal view of entity bias in (large) language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15173–15184, Singapore, December 2023c. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1013. URL `https://aclanthology.org/2023.findings-emnlp.1013`.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. Mitigating label biases for in-context learning. *arXiv preprint arXiv:2305.19148*, 2023.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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)*, Jan 2019. doi: 10.18653/v1/d19-1670. URL `http://dx.doi.org/10.18653/v1/d19-1670`.

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheoneum Park, and Kyomin Jung. Crossaug: A contrastive data augmentation method for debiasing fact verification models. In *Proceedings of the 30th ACM International Conference on Information amp; Knowledge Management*, Oct 2021. doi: 10.1145/3459637.3482078. URL `http://dx.doi.org/10.1145/3459637.3482078`.

Tal Schuster, Darsh J Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. *arXiv preprint arXiv:1908.05267*, 2019.

RabeehKarimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. *arXiv: Computation and Language,arXiv: Computation and Language*, Sep 2019.

Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2):3, 2000.

Judea Pearl. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea Pearl*, pages 373–392. 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.

Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. Debiasing nlu models via causal intervention and counterfactual reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 11376–11384, Jul 2022. doi: 10.1609/aaai.v36i10.21389. URL `http://dx.doi.org/10.1609/aaai.v36i10.21389`.

Wangzhen Guo, Qinkang Gong, and Hanjiang Lai. Counterfactual multihop qa: A cause-effect approach for reducing disconnected reasoning. *arXiv preprint arXiv:2210.07138*, 2022.

Weizhi Xu, Qiang Liu, Shu Wu, and Liang Wang. Counterfactual debiasing for fact verification. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6777–6789, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.374. URL `https://aclanthology.org/2023.acl-long.374`.

Ziwei Chen, Linmei Hu, Weixin Li, Yingxia Shao, and Liqiang Nie. Causal intervention and counterfactual reasoning for multi-modal fake news detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 627–638, 2023a.

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

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammed AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)*, pages 19–30. Association for Computational Linguistics, 2016.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, et al. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, 2021.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Xiaoyu Xing, Zhijing Jin, Di Jin, Bingning Wang, Qi Zhang, and Xuan-Jing Huang. Tasty burgers, soggy fries: Probing aspect robustness in aspect-based sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3594–3605, 2020.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. Understanding and patching compositional reasoning in llms. *arXiv preprint arXiv:2402.14328*, 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

Jun Gao, Wei Wang, Changlong Yu, Huan Zhao, Wilfred Ng, and Ruifeng Xu. Improving event representation via simultaneous weakly supervised contrastive learning and clustering. *arXiv preprint arXiv:2203.07633*, 2022.

Linhai Zhang, Congzhi Zhang, and Deyu Zhou. Multi-relational probabilistic event representation learning via projected gaussian embedding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6162–6174, 2023b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023d. URL `https://openreview.net/forum?id=1PL1NIMMrw`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 126–134, 2005.

Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. Self-evolved diverse data sampling for efficient instruction tuning. *arXiv preprint arXiv:2311.08182*, 2023.

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, 2024.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models. *arXiv preprint arXiv:2305.14264*, 2023.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić, editors, *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL `https://aclanthology.org/2022.deelio-1.10`.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL `https://aclanthology.org/D18-1259`.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 2022.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

AI@Meta. Llama 3 model card. 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

OpenAI. Introducing chatgpt. `https://openai.com/blog/chatgpt`, 2022. Accessed: 2024-02-06.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. Ground-truth labels matter: A deeper look into input-label demonstrations. *arXiv preprint arXiv:2205.12685*, 2022.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L McClelland, Jane X Wang, and Felix Hill. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*, 2022.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2019.

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

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

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

Amir Feder, Katherine A Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E Roberts, et al. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158, 2022.

Ziwei Chen, Linmei Hu, Weixin Li, Yingxia Shao, and Liqiang Nie. Causal intervention and counterfactual reasoning for multi-modal fake news detection. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 627–638, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.37. URL `https://aclanthology.org/2023.acl-long.37`.

Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021a.

Wangzhen Guo, Qinkang Gong, Yanghui Rao, and Hanjiang Lai. Counterfactual multihop QA: A cause-effect approach for reducing disconnected reasoning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4214–4226, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.231. URL `https://aclanthology.org/2023.acl-long.231`.

Ziyang Xu, Keqin Peng, Liang Ding, Dacheng Tao, and Xiliang Lu. Take care of your prompt bias! investigating and mitigating prompt bias in factual knowledge extraction. *arXiv preprint arXiv:2403.09963*, 2024.

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.

Jialong Wu, Linhai Zhang, Deyu Zhou, and Guoqiang Xu. Diner: Debiasing aspect-based sentiment analysis with multi-variable causal inference. *arXiv preprint arXiv:2403.01166*, 2024.

Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. Causal attention for vision-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9847–9857, 2021b.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. CLadder: Assessing causal reasoning in language models. In *NeurIPS*, 2023a. URL `https://openreview.net/forum?id=e2wtjx0Yqu`.

Zhiheng Lyu, Zhijing Jin, Fernando Gonzalez, Rada Mihalcea, Bernhard Schoelkopf, and Mrinmaya Sachan. On the causal nature of sentiment analysis. *arXiv preprint arXiv:2404.11055*, 2024.

Zhijing Jin, Jiarui Liu, LYU Zhiheng, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona T Diab, and Bernhard Schölkopf. Can large language models infer causation from correlation? In *The Twelfth International Conference on Learning Representations*, 2023b.

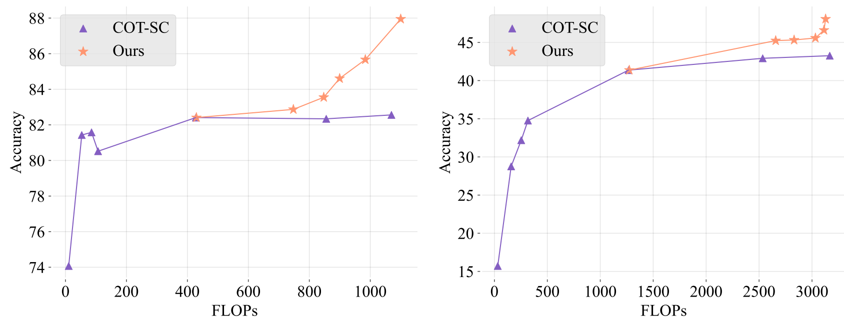
Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schoelkopf, and Mrinmaya Sachan. A causal framework to quantify the robustness of mathematical reasoning with language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 545–561, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.32. URL `https://aclanthology.org/2023.acl-long.32`.

Yujie Lu, Weixi Feng, Wanrong Zhu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. Neuro-symbolic procedural planning with commonsense prompting. In *The Eleventh International Conference on Learning Representations*, 2022.

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.

Sara Abdali, Anjali Parikh, Steve Lim, and Emre Kiciman. Extracting self-consistent causal insights from users feedback with llms and in-context learning. *arXiv preprint arXiv:2312.06820*, 2023.

Appendix Contents



(a) GSM8K (b) MuSiQue

Figure 5: Comparison of FLOPs cost between Causal Prompting and CoT-SC method on LLaMA3.

A Discussion

In this section, we will address the following Research Questions (**RQ**) to elucidate our contributions more clearly.

RQ1: How can we reduce computational costs incurred by additional operations?
Compared to CoT-SC, Causal Prompting involves numerous additional operations, resulting in increased cost overhead. Importantly, **not** every test instance necessitates a front-door adjustment. To determine whether a a front-door adjustment is needed for a given example, we employ a *self-consistency metric*, defined as the proportion of answers receiving the majority of votes. A front-door adjustment is executed if the *self-consistency metric* falls below a predefined threshold s. As shown in Figure 5, adjusting the threshold s allows us to control the cost of the entire test dataset. In our experiments, the computational cost is quantified in terms of Floating Point Operations per Second (FLOPs). For a test set containing 1000 samples with an average sample length of 500, the inference time on LLaMA3 is approximately two hours using the vLLM framework [Kwon et al., 2023]. All computations using open-source LLMs were executed on an NVIDIA A100 80GB GPU. The value of the threshold s ranges from 0 to 1; a smaller s results in more samples requiring front-door adjustment and consequently increases the cost of FLOPs. To facilitate a fair comparison of equivalent costs, we set the number of votes for CoT-SC at $m = 1, 5, 8, 10, 40, 80, 100$. Our method achieves **superior** performance at equivalent costs. As computational costs increase, the performance of CoT-SC gradually reaches its upper limit, whereas the performance of our proposed method continues to rise. It indicates Causal Prompting has more potential to scale effectively with increased computational costs.

RQ2: How does adjusting the threshold s affect performance?
We also explored the effect of the threshold s on different tasks on LLaMA3. As shown in Figure 6, the performance of mathematical reasoning and multi-hop question answering tasks keeps improving as the threshold s increases. On the other hand, the performance on the NLU task first increases and then decreases, and when we apply the front-door adjustment to all samples (*i.e.*, $s = 1.0$), the performance drops significantly. This is because we introduce the wrong and correct chain-of-thoughts in the NWGM algorithm part, and the labels of the two chain-of-thoughts are usually opposite, which is easy to cause LLM to change the correct answer into the wrong one, especially in classification tasks such as NLU. The best threshold s across different backbone LLMs and different datasets are shown in Table 3. Table 1 reports the model's performance when the threshold s takes the best value.

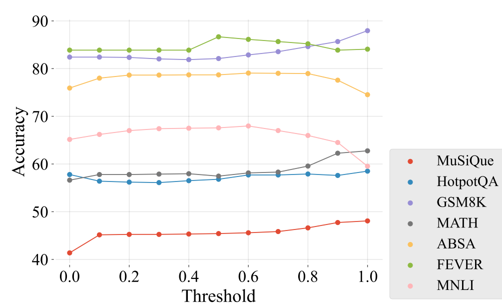


Figure 6: The impact of threshold s on LLaMA3-8B.

RQ3: What are the effects of employing contrastive learning methods?
To better explore the importance of contrastive learning, we present a visualization of the embeddings of the chain-of-thought obtained by the vanilla $\mathrm{Encoder}$ and the $\mathrm{Encoder}$ trained by contrastive learning with T-SNE [Van der Maaten and Hinton, 2008]. As shown in Figure 7, the vanilla $\mathrm{Encoder}$ fails to distinctly separate different categories, blending multiple chain-of-thoughts into indistinct representations. In contrast, the $\mathrm{Encoder}$ that is trained by contrastive learning exhibits a clear delineation between different categories.

Table 3: The best threshold s across different backbone LLMs, including LLaMA2, LLaMA3 and GPT-3.5, on seven datasets.

Threshold s	**GSM8K**	**MATH**	**HotpotQA**	**MuSiQue**	**ABSA**	**NLI**	**FV**
LLaMA2	1.0	0.2	0.5	1.0	0.4	0.4	0.6
LLaMA3	1.0	1.0	1.0	1.0	0.6	0.6	0.5
GPT-3.5	1.0	0.3	1.0	0.1	0.6	0.9	1.0

Table 4: Notations used in our proposed method **Causal Prompting**.

Notation	**Description**
$\mathrm{Encoder}$	The encoder-only model for generating text embedding
LLM	The large language model for text generation
$Sort$	The function to sort the training set
$\mathcal{D}$	The training set
d	The demonstration examples in the prompt
q^{test}	The test sample
n	The number of demonstration examples in the prompt
m	The number of CoTs generated by LLM
K	The number of clusters for the K-means clustering algorithm
T	The query times to LLM for each intervention prompt

This shows that contrastive learning can align the representation spaces of the encoder and LLMs, allowing the encoder to learn how to distinguish the semantics between different chain-of-thoughts generated by the LLMs. In Appendix E.2, we present quantitative analyses to further substantiate the effectiveness of contrastive learning.

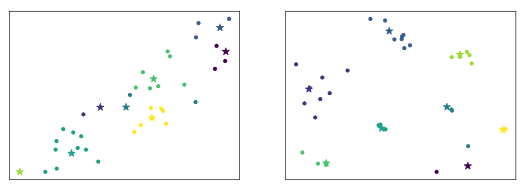


Figure 7: Visualization of the embeddings of chain-of-thought obtained by the original $\mathrm{Encoder}$ (left) and the $\mathrm{Encoder}$ trained through contrastive learning (right) on the GSM8K dataset.

RQ4: Why do we not need any additional baselines?

See Appendix C.1, where we list all the baselines used to compare with our proposed method. However, numerous other prompting strategies such as ToT [Yao et al., 2024] or GoT [Besta et al., 2024] have not been explored as baselines. We do not conclude these strategies based on the following three primary considerations: **First**, we focus on solving the more fundamental problem of prompting methods: the estimation of the causal effect between the input prompt and the response of the LLM. The efficacy of our causal prompting approach is substantiated by the experimental results presented. **Second**, more complex prompting methods require the construction of intricate causal graphs. In other words, building causal graphs for these advanced prompting methods is challenging due to their involvement with cycles and numerous mediating variables. As an early work on combining causal inference with LLM, we start with the basic prompting method. **Finally**, our approach is not specifically intended to improve the ability of LLMs to solve complex problems, but rather to hope that LLMs generate *more causal, faithful and unbiased* answers. Therefore, we only experimented on the most commonly used tasks and did not compare them with other prompting methods designed for complex tasks.

B Causal Prompting Algorithm

B.1 Notations

The notations used in the approach are shown in Table 4 to clarify their usage and significance throughout the algorithm.

B.2 Algorithm Details

As shown in Algorithm 1, we describe the operation flow of Causal Prompting.

Algorithm 1 Causal Prompting

Input: $\text{Encoder}, \text{LLM}, Sort, \mathcal{D}, d, q^{test}, n, m, K, T$

$\mathcal{P} \leftarrow [d_1, ..., d_n, q^{test}]$
$\{c_i | i = 1, ..., m\} \leftarrow \text{LLM}(\mathcal{P})$
$\overline{c}_i \leftarrow \text{Encoder}(\text{[CLS]}, c_i, \text{[SEP]})$
$\{C_1, ..., C_K\} \leftarrow \text{K-means}(\overline{c}_1, ..., \overline{c}_m)$
for $k = 1$ **to** K:
  $r_k \leftarrow \text{Center}(C_k)$
  $P(r_k | do(X)) \leftarrow \frac{|C_k|}{m}$
end for
for $k = 1$ **to** K:
  $\overline{r}_k \leftarrow \text{Encoder}(\text{[SEP]}, r_k, \text{[SEP]})$
  $\overline{d}_j \leftarrow \text{Encoder}(\text{[CLS]}, r_j^{wrong}, \text{[SEP]})$
  $\{d_j^{\uparrow}\}_{j=1}^{N} \leftarrow Sort(\mathcal{D}, \overline{r}_k, \{\overline{d}_j\}_{j=1}^{N})$
  $\mathcal{P}_{r_k}^{iter} \leftarrow [d_l^{\uparrow}, ..., d_1^{\uparrow}, q^{test}]$
  $\{(r_{k,t}^{ip}, a_{k,t}) | t = 1, ..., T\} \leftarrow \text{LLM}(\mathcal{P}_{r_k}^{iter}, r_k)$
  $P(A | do(r_k)) \leftarrow \frac{\sum_{t=1}^{T} \mathbb{I}(A = a_{k,t})}{T}$
end for
$P(A | do(X)) \leftarrow \sum_{k=1}^{K} \frac{|C_k|}{m} \cdot \frac{\sum_{t=1}^{T} \mathbb{I}(A = a_{k,t})}{T}$
return $argmax_A(P(A | do(X)))$

C Experimental Details

C.1 Baselines

Standard ICL [Brown et al., 2020]: Prompt LLMs with some demonstration examples containing only questions and their corresponding answers, without any additional explanatory context or reasoning.

CoT [Wei et al., 2022]: Unlike Standard ICL, CoT method enhances the prompt with demonstration examples that include detailed chain-of-thoughts. These chain-of-thoughts guide the LLMs through the steps required to reach an answer.

CoT-SC [Wang et al., 2022]: Extent the CoT methods by having the LLMs generate multiple different chain-of-thoughts for the same query and use majority voting to determine the final answer.

C.2 Settings

Demonstration Construction For the GSM8K and MATH datasets, we directly use the gold rationale in the dataset as the chain-of-thought for the demonstration. For multi-hop question answering and NLU tasks without gold rationale, we use a few manually constructed demonstrations to prompt the LLM to generate chain-of-thoughts and answers for all examples in the dataset. For the samples with wrong answers, we provide the LLM with the correct answers and then ask the LLM to generate the correct chain-of-thoughts. See Appendix H.4 for the prompt template used to generate the demonstration examples. Finally, we retain the wrong and correct chain-of-thoughts and use them to construct the training set. Note that to better evaluate the debiasing effect of our method, we only use the original dataset to build demonstration examples without including the adversarial dataset, and evaluate on both original and adversarial datasets.

Demonstration Selection To fair comparison, the same demonstration samples are utilized across all few-shot prompting methods. For each instance, the most relevant demonstration samples are selected based on the similarity of their embeddings with the question. These selected demonstration samples are then concatenated into the prompt, as exemplified in the Appendix H.1 for CoT prompting. Specifically, for Math Reasoning and Multi-hop QA tasks, 4-shot and 2-shot settings are employed, respectively, corresponding to $n = 4$ and $n = 2$ in Equation (6). For NLU tasks, we maintain a balanced label space by including one demonstration sample for each category. For ABSA and NLI, which are 3-way classification tasks, $n = 3$ is adopted, while for FV, a 2-way task, $n = 2$ is used. After the application of NWGM-based causal intervention, as described in Section 3.2, and l different demonstration samples are selected to be incorporated into the prompt as shown in the prompt templates of CoT Improvement based on NWGM approximation in Appendix H.2. For the mathematical reasoning task, $l = 2$ in Equation (16). For the multi-hop question answering task, $l = 1$. For the ABSA and NLI tasks, $l = 3$. For the FV task, we set $l = 2$.

C.3 Implementation Details

Details of LLM We evaluate our prompting method on the open-source LLaMA-2-7B-Chat [Touvron et al., 2023][2] and LLaMA-3-8B-Instruct [AI@Meta, 2024][3] using *Transformer*[Wolf et al., 2019] library, as well as the closed-source `GPT-3.5-turbo-0125` [OpenAI, 2022][4]. The generation hyperparameters remain consistent across all prompting methods: temperature is set to 0.7, and top_p is set to 0.9. Following previous work [Lyu et al., 2023], we set the number of votes in COT-SC is 40. In our method, the number of chain-of-thoughts generated in the first part is $m = 40$, and then these chain-of-thoughts are clustered into $K = 8$. For each chain-of-thought representing the cluster center, we generated $T = 10$ answers based on the prompt modified by intervention. Finally, the $K \cdot T = 80$ answers were weighted voting to get the final answer.

Details of $\mathrm{Encoder}$ We use BERT-base [Devlin et al., 2018][5] as the $\mathrm{Encoder}$ in computing sentence similarity, clustering algorithm, and NWGM algorithm following [Tian et al., 2022, Zhang et al., 2024]. We independently fine-tune an $\mathrm{Encoder}$ for each LLM, as well as for each specific task, employing a contrastive learning approach. During training, we set the batch size is 128. The learning rate is $1e - 4$. The temperature $temp$ is set to 0.3. The max length of $\mathrm{Encoder}$ is 512. The total training epochs are 20.

C.4 Dataset Details

Math Reasoning For the GSM8K [Cobbe et al., 2021] dataset, we use its official dataset split[6]. The number of samples in the training set is 7473 and the number of samples in the test set is 1319. For the MATH [Hendrycks et al., 2021] dataset, we only use algebra type data due to limited computational resources. We use the official dataset split[7], the training set of MATH-algebra includes 1744 samples, and the test set includes 1187 samples.

Multi-hop Question Answering For the HotpotQA [Yang et al., 2018] dataset, to reduce the experimental cost, we randomly selected 5000 samples from the official training set[8] as our training set and 1000 samples from the official validation set[9] as our test set. For the MuSiQue [Trivedi et al., 2022] dataset, we use the official dataset split of MuSiQue-Answerable version[10], and we experiment on the more challenging part of the dataset with $hop \geq 3$. After extracting the data with a $hop \geq 3$, the number of training sets is 5562 and the number of test sets is 1165. For both datasets above, we extract the support documents to form the paragraph.

Natural Language Understanding For the Aspect-based Sentiment Analysis (ABSA) and Natural Language Inference (NLI) tasks, we use SemEval2014-Laptop [Pontiki et al., 2016] and MNLI-m [Williams et al., 2017] as the original datasets (in-distribution, ID) and the corresponding transformation data generated by TextFlint [Wang et al., 2021] as the adversarial datasets (out-of-distribution, OOD). For the FV task, we use FEVER [Thorne et al., 2018] as the ID dataset and its adversarial dataset Symmetric FEVER [Schuster et al., 2019] as the OOD dataset. To reduce the experimental cost, we randomly sample a certain number of samples from these NLU datasets for experiments. See Table 5 and Table 6 for details of the adversarial dataset generation method and data statistics for the NLU dataset.

C.5 Evaluation

Following previous works [Ye et al., 2023, Lyu et al., 2023], for Math Reasoning and NLU tasks, we adopt the label classification accuracy (Acc) as the evaluation metric, and for Multi-hop QA tasks, we adopt the Exact Match (EM) and F1 [Yang et al., 2018] as the evaluation metric. Following previous work [Lyu et al., 2023], we extract the text span following the keyword "*answer is*" as the answer.

D Related Works

D.1 Prompting Strategies

The performance of LLMs on downstream tasks is largely influenced by the employed prompting strategy [Sahoo et al., 2024]. Current prompting strategies primarily improve the quantity and robustness through two approaches.

[2]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[3]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[4]https://platform.openai.com/docs/models/gpt-3-5-turbo
[5]https://huggingface.co/google-bert/bert-base-uncased
[6]https://github.com/openai/grade-school-math
[7]https://github.com/hendrycks/math
[8]http://curtis.ml.cmu.edu/datasets/hotpot/hotpot_train_v1.1.json
[9]http://curtis.ml.cmu.edu/datasets/hotpot/hotpot_dev_distractor_v1.json
[10]https://github.com/StonyBrookNLP/musique

Table 5: Multiple adversarial categories for ABSA, NLI, and FV tasks.

Task	Adversarial Category	Description
ABSA	ReverseTarget	Reverse the sentiment of the target aspect.
	ReverseNonTarget	Reverse the sentiment of the non-target aspects with originally the same sentiment as target.
	AddDiff	Add aspects with the opposite sentiment from the target aspect.
NLI	AddSent	Add some meaningless sentence to premise, which do not change the semantics.
	NumWord	Find some num words in sentences and replace them with different num word.
	SwapAnt	Find some keywords in sentences and replace them with their antonym.
FV	Symmetric	For each claim-evidence pair, generating a synthetic pair that holds the same relation (e.g. SUPPORTS or REFUTES) but expressing a different, contrary, fact.

Table 6: Details of all datasets used in the experiments.

Tasks	Datasets	Train	Test		Measure
			`Ori`	`Adv`	
Math Reasoning	GSM8K	7473	1319	-	Accuracy
	MATH	1744	1187	-	Accuracy
Multi-hop Question Answering	HotpotQA	5000	1000	-	F1 & EM
	MuSiQue	5562	1165	-	F1 & EM
Natural Language Understanding	ABSA	2358	638	1239	Accuracy
	NLI	5000	810	754	Accuracy
	FV	5000	500	1000	Accuracy

On one hand, incorporating a few labeled examples within the instruction can significantly enhance the performance of LLM, occasionally even better than fine-tuned models [Brown et al., 2020, Chung et al., 2022, Dong et al., 2022]. This method is referred to as In-Context Learning (ICL). Several works involve the modification of instruction examples to address the issue of LLMs being particularly sensitive to the designing of demonstration examples, including example selection [Liu et al., 2021], example format Dong et al. [2022], example label [Min et al., 2022, Yoo et al., 2022], and example order [Lu et al., 2021]. Following this way, recent works have suggested that including chain-of-thoughts (CoTs) in the context of these examples can further augment the quality of LLM's responses [Nye et al., 2021, Lampinen et al., 2022, Wei et al., 2022]. **On the other hand**, to tackle the instability in the outputs of LLMs, attributable to the *"multinomial sampling"* [Holtzman et al., 2019], some works involve enhancement by employing multiple sampling and voting mechanisms to determine the final answer [Chen et al., 2021, Wang et al., 2022, Li et al., 2022], which is referred to as self-consistency (SC). Moreover, for more complex tasks, many other prompting methods based on multiple sampling have been proposed, such as React [Yao et al., 2022], ToT [Yao et al., 2024], and GoT [Besta et al., 2024].

In this work, we primarily combine these two directions by modeling front-door adjustments to address bias issues in LLM prompting. CoT is utilized as a mediator variable. In such scenarios, we further employ NWGM approximation to select the demonstration, which can represent the expectation of the entire dataset and help the model generate an unbiased answer.

D.2 Debiasing with Causal Inference

Causal inference uses scientific methods to identify causal relationships between variables [Pearl et al., 2016]. Because of its rigorous theoretical guarantees and mature causal modeling tools [Pearl, 2019], causal inference has advantages in debiasing work. Recently, causal inference has been widely used in natural language processing [Feder et al., 2022, Chen et al., 2023b] and computer vision [Yang et al., 2021a].

Some works apply counterfactual reference to remove the bias of the model [Xu et al., 2023, Guo et al., 2023, Niu et al., 2021, Xu et al., 2024]. For instance, Niu et al. [2021] debias the visual question answering task by subtracting the predictions of the language-only model from the predictions of the vision-language model to reduce linguistic biases in the integrated system. Other work uses causal interventions for debiasing, including backdoor adjustment [Tian et al., 2022, Zhu et al., 2023, Wang et al., 2023c, Wu et al., 2024] and front-door adjustment [Yang et al., 2021b,a, Zhang et al., 2024]. In the era of LLMs, several studies have integrated

Table 7: Results of the robustness study on LLaMA2 and GPT-3.5. `Ori` denotes the original dataset (in-distribution) and `Adv` denotes the adversarial dataset (out-of-distribution). The best results are in **bold**.

	ABSA		NLI		FV	
Methods	`Ori`	`Adv`	`Ori`	`Adv`	`Ori`	`Adv`
			LLaMA2			
Standard ICL	71.63	34.71	36.42	19.36	72.00	49.30
CoT	66.14	40.36	29.51	25.46	76.60	66.80
CoT-SC	**75.24**	42.62	36.91	29.97	77.80	69.40
Causal Prompting	73.04	**64.73**	**55.06**	**46.29**	**89.00**	**77.10**
			GPT-3.5			
Standard ICL	75.24	66.18	74.32	31.17	88.20	68.90
CoT	70.69	63.2	74.07	52.25	89.60	76.20
CoT-SC	79.94	71.83	76.91	56.10	91.00	78.60
Causal Prompting	**82.29**	**79.02**	**80.74**	**62.47**	**94.60**	**82.50**

Table 8: The results of ablation study on LLaMA3. The best results are in **bold**.

	GSM8K	MATH	HotpotQA		MuSiQue	
Methods	Acc	Acc	EM	F1	EM	F1
Causal Prompting	**87.95**	**62.76**	**58.5**	**78.18**	**48.07**	**64.23**
NWGM-`Reverse`	87.72	62.26	57.7	77.97	47.81	63.48
NWGM-`Random`	86.13	52.99	56.9	77.23	47.38	61.60
w/o Contrastive Learning	86.58	59.56	57.9	77.88	47.47	62.45

LLMs with causal inference techniques [Jin et al., 2023a, Lyu et al., 2024, Jin et al., 2023b, Stolfo et al., 2023]. However, some of these studies do not fully adhere to the structural causal model, relying excessively on heuristic methods [Lu et al., 2022, Wang et al., 2023c, Zhang et al., 2023a, Tang et al., 2023]; others employ overly simplistic causal diagrams, which are inadequate for complex tasks [Abdali et al., 2023, Lyu et al., 2024]. Overall, counterfactual inference necessitates the acquisition of logits from LLM outputs, whereas back-door adjustment demands modeling of specific values of confounding variables. In contrast, front-door adjustment allows causal intervention without access to the values of confounding variables or logits of LLM outputs. This makes front-door adjustment particularly apt for application in LLM scenarios.

Therefore, we propose to debias the prompting methods by causal intervention based on front-door adjustment. The work most closely related to ours is DeCOT [Lanham et al.]; DeCoT [Lanham et al.] debiases the chain-of-thoughts of LLMs by incoroporating counterfactual knowledge and front-door adjustment. Both our work and DeCoT use the front-door adjustment on LLMs. However, DeCoT requires the introduction of instrumental variables to model counterfactual knowledge, limiting its applicability to knowledge-intensive tasks. In contrast, our approach is versatile and can be applied to a wide range of tasks. Our approach, on one hand, adapts the traditional front-door adjustment to make it suitable for the task of LLM prompting. On the other hand, it adheres more closely to the established principles of the field.

E More Experimental Results

E.1 Robustness Study

We also provide robustness studies on LLaMA2 and GPT3.5 in Table 7. The findings are consistent with experiments conducted on LLaMA3 in Section 4.4, demonstrating that our method possesses a distinct advantage on adversarial robustness datasets.

E.2 Ablation Study

In the ablation study presented in Table 8, we perform a detailed ablation analysis to evaluate two pivotal aspects: the effectiveness of the NWGM approximation and the impact of incorporating contrastive learning. This analysis spans four datasets, including GSM8K, MATH, HotpotQA, and MuSiQue, utilizing the LLaMA3-8B model.

Effectiveness of the NWGM The NWGM approximation is employed in the Estimation of $P(A|do(r))$ to perform the back-door adjustment, where we select the ICL demonstration that is most similar to the query and put the most similar demonstration samples closer to the test samples, as detailed in Section 3.2. We evaluate the impact of the NWGM approximation by comparing the standard setup against two variants of NWGM-`Reverse` and NWGM-`Random`, respectively. NWGM-`Reverse` means that we reverse the order of standard ICL demonstrations, that is, $\mathcal{P}_{r_k}^{iter} = [d_1^{\uparrow}, ..., d_n^{\uparrow}, q^{test}]$ with the same order of KATE [Liu et al., 2021] method. NWGM-`Random` denotes that ICL demonstrations are selected at random from the raining set $\mathcal{D}$. The performance decline observed in these two variants validates the effectiveness of our approach in selecting the most relevant samples, thereby enabling a more accurate estimation of $P(A|do(r))$.

Impact of contrastive learning The second aspect of our ablation study is to assess the role of contrastive learning. By removing contrastive learning, we can observe a decline in performance metrics compared to Causal Prompting. For example, there is a decrease of 1.37% in accuracy on GSM8K and 3.20% on MATH. It indicates that contrastive learning facilitates the alignment of feature representations between the $Encoder$ and LLMs.

F Limitations

Although our results already outperform baselines overall, our work still suffers from the following limitations.

- We evaluate the effectiveness of our approach on three tasks: Math Reasoning, Multi-hop Question Answering, and Natural Language Understanding. We need to test the effectiveness of Causal Prompting on other more complex tasks.
- As mentioned in Appendix A, although our method outperforms baselines at the same cost, how to reduce the cost of prompt remains an important issue.
- We only evaluated the effectiveness of Causal Prompting on three Large Language Models, LLaMA2-7B, LLaMA3-8B and GPT-3.5, and we need to evaluate our method on more Large Language Models of different kinds and scales.

G Case Study

In this section, we provide two running samples from GSM8K and HotpotQA with intermediate output for each module.

G.1 Case on GSM8K

Intermediate output details of a GSM8K example.

Question

A merchant wants to make a choice of purchase between 2 purchase plans: jewelry worth $5,000 or electronic gadgets worth $8,000. His financial advisor speculates that the jewelry market will go up 2.5% while the electronic gadgets market will rise 1.2% within the same month. If the merchant is looking to maximize profit at the end of this month by making a choice, how much profit would this be?
(The correct answer is 125)

Outputs from CoT-SC

The number of chain−of−thoughts m=40.
Answer voting results:
$P(A = 96|X) = 19/40 = 0.475$
$P(A = 125|X) = 18/40 = 0.45$
...
(Omit answers with fewer votes)
Therefore, the final answer obtained according to the COT−SC method is 96.

An example of an error is shown:

Let's calculate the future value of each option:
Option 1: Jewelry
Current value = $5,000
Expected increase = 2.5% of $5,000 = $125
Future value = $5,000 + $125 = $5,125

Option 2: Electronic Gadgets
Current value = $8,000
Expected increase = 1.2% of $8,000 = $96
Future value = $8,000 + $96 = $8,096
Since the merchant wants to maximize profit, they should choose the option with the higher future value. In this case, it's Option 2: Electronic Gadgets with a future value of $8,096.
The profit would be the difference between the future value and the current value: $8,096 − $8,000 = $96.
Therefore, the answer is $96.

Cluster results

The number of chain−of−thoughts selected from clustering results k=8.
——————————————
CoT−1: $P(r_1|X) = 0.375$
——————————————
Let's calculate the profit for each option:
Option 1: Jewelry
∗ Current value: $5,000
∗ Expected increase: 2.5% of $5,000 = $125
∗ Future value: $5,000 + $125 = $5,125
∗ Profit: $5,125 − $5,000 = $125
Option 2: Electronic Gadgets
∗ Current value: $8,000
∗ Expected increase: 1.2% of $8,000 = $96
∗ Future value: $8,000 + $96 = $8,096
∗ Profit: $8,096 − $8,000 = $96
Comparing the profits, the merchant would make a higher profit by choosing the jewelry option, which is $125. Therefore, the answer is $125.

——————————————
CoT−2: $P(r_2|X) = 0.025$
——————————————
Let's analyze the situation step by step:
1. The merchant has to choose between jewelry worth $5,000 and electronic gadgets worth $8,000.
2. The financial advisor predicts that the jewelry market will go up 2.5% within the same month. This means that the value of the jewelry will increase by 2.5% of $5,000, which is:
$5,000 x 0.025 = $125
So, the value of the jewelry will become:
$5,000 + $125 = $5,125
3. The financial advisor predicts that the electronic gadgets market will rise 1.2% within the same month. This means that the value of the electronic gadgets will increase by 1.2% of $8,000, which is:
$8,000 x 0.012 = $96
So, the value of the electronic gadgets will become:
$8,000 + $96 = $8,096
4. Now, let's compare the profit from each option:
Option 1: Jewelry ($5,125 − $5,000) = $125
Option 2: Electronic Gadgets ($8,096 − $8,000) = $96
Since $125 is greater than $96, the merchant would make a maximum profit by choosing the jewelry option, which is $125.

——————————————
CoT−3: $P(r_3|X) = 0.075$
——————————————
Let's analyze the situation:
Option 1: Jewelry worth $5,000
∗ The jewelry market will go up 2.5%, so the value will increase by 2.5% of $5,000 = $125
∗ The merchant will sell the jewelry for $5,000 + $125 = $5,125
∗ The profit = Selling price − Cost price = $5,125 − $5,000 = $125
Option 2: Electronic gadgets worth $8,000
∗ The electronic gadgets market will go up 1.2%, so the value will increase by 1.2% of $8,000 = $96
∗ The merchant will sell the electronic gadgets for $8,000 + $96 = $8,096
∗ The profit = Selling price − Cost price = $8,096 − $8,000 = $96

Comparing the profits, we can see that the merchant would make a profit of $125 if they choose the jewelry option, and a profit of $96 if they choose the electronic gadgets option. Since $125 is greater than $96, the merchant would maximize their profit by choosing the jewelry option.

——————————————
CoT–4: $P(r_4|X) = 0.025$
——————————————

Let's analyze the situation step by step:
1. If the merchant chooses to buy jewelry worth $5,000, its value will increase by 2.5% in the next month. To find the increase in value, multiply the current value by the percentage increase:
$5,000 x 2.5% = $5,000 x 0.025 = $125
The total value of the jewelry after the increase will be:
$5,000 + $125 = $5,125
2. If the merchant chooses to buy electronic gadgets worth $8,000, its value will increase by 1.2% in the next month. To find the increase in value, multiply the current value by the percentage increase:
$8,000 x 1.2% = $8,000 x 0.012 = $96
The total value of the electronic gadgets after the increase will be:
$8,000 + $96 = $8,096
3. To maximize profit, the merchant should choose the option that results in the highest total value. In this case, the merchant should choose to buy electronic gadgets worth $8,096, which is $976 more than the value of the jewelry.
The profit will be the difference between the increased value of the electronic gadgets and their original price:
$8,096 – $8,000 = $96
So, the maximum profit the merchant can make is $96.

——————————————
CoT–5: $P(r_5|X) = 0.15$
——————————————

Let's calculate the value of the jewelry after the market goes up 2.5%:
Initial value: $5,000
Percentage increase: 2.5%
Value after increase: $5,000 + (2.5% of $5,000) = $5,000 + $125 = $5,125
Now, let's calculate the value of the electronic gadgets after the market goes up 1.2%:
Initial value: $8,000
Percentage increase: 1.2%
Value after increase: $8,000 + (1.2% of $8,000) = $8,000 + $96 = $8,096
Since the merchant wants to maximize profit, he should choose the option that gives him the highest value after the market increase. In this case, the electronic gadgets option is worth $8,096, which is higher than the jewelry option worth $5,125.
The profit would be the difference between the value of the chosen option and its initial value:
Profit = $8,096 – $8,000 = $96
Therefore, the answer is $96.

——————————————
CoT–6: $P(r_6|X) = 0.025$
——————————————

Let's analyze the situation step by step:
1. The merchant has two options: purchasing jewelry worth $5,000 or electronic gadgets worth $8 ,000.
2. The financial advisor predicts that the jewelry market will rise by 2.5%, which means the value of the jewelry will increase by 2.5% of $5,000 = $125.
3. The value of the jewelry will then become $5,000 + $125 = $5,125.
4. The financial advisor also predicts that the electronic gadgets market will rise by 1.2%, which means the value of the electronic gadgets will increase by 1.2% of $8,000 = $96.
5. The value of the electronic gadgets will then become $8,000 + $96 = $8,096.
6. The merchant wants to maximize profit, which means they should choose the option that results in the highest profit.
Comparing the two options:
∗ Jewelry: $5,125 (new value) – $5,000 (original value) = $125 profit
∗ Electronic gadgets: $8,096 (new value) – $8,000 (original value) = $96 profit
The merchant should choose the jewelry option, as it results in a higher profit of $125 compared to $96 for the electronic gadgets. Therefore, the answer is $125.

——————————
CoT–7: $P(r_7|X) = 0.15$
——————————
Let's calculate the profit for each option:
Option 1: Jewelry worth $5,000
The jewelry market will rise 2.5%, so the value of the jewelry will increase to:
$5,000 x (1 + 0.025) = $5,000 x 1.025 = $5,125
Profit = Selling price − Purchase price = $5,125 − $5,000 = $125
Option 2: Electronic gadgets worth $8,000
The electronic gadgets market will rise 1.2%, so the value of the gadgets will increase to:
$8,000 x (1 + 0.012) = $8,000 x 1.012 = $8,096
Profit = Selling price − Purchase price = $8,096 − $8,000 = $96
Since the merchant is looking to maximize profit, they should choose Option 1, which yields a profit of $125.

——————————
CoT–8: $P(r_8|X) = 0.175$
——————————
Let's analyze the situation step by step:
1. Jewelry worth $5,000 with a 2.5% increase:
New value = $5,000 + (2.5% of $5,000) = $5,000 + $125 = $5,125
Profit = New value − Original value = $5,125 − $5,000 = $125
2. Electronic gadgets worth $8,000 with a 1.2% increase:
New value = $8,000 + (1.2% of $8,000) = $8,000 + $96 = $8,096
Profit = New value − Original value = $8,096 − $8,000 = $96
Comparing the profits, the merchant would make a profit of $125 if they choose the jewelry plan and a profit of $96 if they choose the electronic gadgets plan. Since the merchant wants to maximize profit, they should choose the jewelry plan, which would yield a profit of $125.

Intervention results

For the above eight chain–of–thoughts, we use the NWGM algorithm to perform causal intervention operations on them respectively. We can then compute the causal effect between the chain–of–thought and the answer.

$P(A = 125|do(r_1)) = 9/10 = 0.9$
$P(A = 96|do(r_1)) = 0/10 = 0.0$

$P(A = 125|do(r_2)) = 10/10 = 1.0$
$P(A = 96|do(r_2)) = 0/10 = 0.0$

$P(A = 125|do(r_3)) = 10/10 = 1.0$
$P(A = 96|do(r_3)) = 0/10 = 0.0$

$P(A = 125|do(r_4)) = 0/10 = 0.0$
$P(A = 96|do(r_4)) = 8/10 = 0.8$

$P(A = 125|do(r_5)) = 0/10 = 0.0$
$P(A = 96|do(r_5)) = 10/10 = 1.0$

$P(A = 125|do(r_6)) = 9/10 = 0.9$
$P(A = 96|do(r_6)) = 1/10 = 0.1$

$P(A = 125|do(r_7)) = 8/10 = 0.8$
$P(A = 96|do(r_7)) = 0/10 = 0.0$

$P(A = 125|do(r_8)) = 10/10 = 1.0$
$P(A = 96|do(r_8)) = 0/10 = 0.0$

Final results

The final answer is obtained by performing a weighted voting as follows:
$P(A = 125|do(X))$ = 0.375 $*$ 0.9 + 0.025 $*$ 1.0 + 0.075 $*$ 1.0 + 0.025 $*$ 0.0 + 0.15 $*$ 0.0 + 0.025 $*$ 0.9 + 0.15 $*$ 0.8 + 0.175 $*$ 1.0 = 0.755

$P(A = 96|do(X))$ = 0.375 $*$ 0.0 + 0.025 $*$ 0.0 + 0.075 $*$ 0.0 + 0.025 $*$ 0.8 + 0.15 $*$ 1.0 + 0.025 $*$ 0.1 + 0.15 $*$ 0.0 + 0.175 $*$ 0.0 = 0.1725
Finally, we chose the answer with the largest weight as the final answer.
Therefore, the final answer obtained according to the Causal Prompting method is 125.

G.2 Case on HotpotQA

Intermediate output details of a HotpotQA example.

Context

The Walton family is an American family whose collective fortune makes them among the richest families in the world. The majority of their wealth derives from the heritage of Bud and Sam Walton, who were the co–founders of the world's largest retailer, Walmart. The three most prominent living members (Jim, Rob and Alice) have consistently been in the top ten of the "Forbes" 400 since 2001, as were John (2005) and Helen (d. 2007) prior to their deaths. Christy Walton took her husband John 's place after his death. Helen Robson Kemper Walton (December 3, 1919 – April 19, 2007) was the wife of Wal–Mart and Sam's Club founder Sam Walton. At one point in her life, she was the richest American and the eleventh–richest woman in the world.

Question

Who was the husband of the prominent Walton family member who died after John died in 2005?
(The correct answer is Sam Walton)

Outputs from CoT-SC

The number of chain–of–thoughts m=40.
Answer voting results:
$P(A = John|X) = 13/40 = 0.325$
$P(A = SamWalton|X) = 12/40 = 0.3$
$P(A = JohnWalton|X) = 9/40 = 0.225$
...
(Omit answers with fewer votes)
Therefore, the final answer obtained according to the COT–SC method is John.

An example of an error is shown:

Step 1: Identify the prominent Walton family member who died after John died in 2005. Helen Robson Kemper Walton died in 2007.
Step 2: Determine who Helen was married to. Helen was the wife of Sam Walton.
Step 3: Identify who took Helen's place after her death. Christy Walton took her husband John's place after his death.
Step 4: Determine who John is. John is the husband of Christy Walton.
Step 5: Combine the information to answer the question. The husband of the prominent Walton family member who died after John died in 2005 is John.
Therefore, the answer is: John.

Cluster results

The number of chain–of–thoughts selected from clustering results k=8.
————————————————
CoT–1: $P(r_1|X) = 0.325$
————————————————
Step 1: Identify the prominent Walton family member who died after John died in 2005. According to the text, Helen Robson Kemper Walton died in 2007.
Step 2: Determine who Helen's husband was. According to the text, Helen was the wife of Sam Walton.
Step 3: Determine who John's wife was. According to the text, John's wife was Christy Walton.
Step 4: Combine the information to answer the question. Since Helen was Sam's wife and Christy is John's wife, it means that John is not Helen's husband. Therefore, the answer is: Sam Walton.

————————————————
CoT–2: $P(r_2|X) = 0.25$
————————————————

Step 1: Identify the prominent Walton family member who died after John died in 2005. The prominent Walton family member who died after John died in 2005 is Helen (d. 2007).
Step 2: Determine who Helen was married to. Helen was married to Sam Walton, the founder of Walmart and Sam's Club.
Step 3: Identify the current spouse of Helen. Christy Walton took her husband John's place after his death.
Step 4: Determine who John is. John is Christy Walton's husband.
Step 5: Combine the information to answer the question. Helen's husband after John died in 2005 is John, who is also Christy Walton's husband.
Therefore, the answer is: John.

——————————————
CoT–3: $P(r_3|X) = 0.075$
——————————————

Step 1: Identify the prominent Walton family member who died after John died in 2005. The prominent Walton family member who died after John died in 2005 is Helen Robson Kemper Walton.

Step 2: Determine the husband of Helen Robson Kemper Walton. Helen Robson Kemper Walton was the wife of Sam Walton.
Step 3: Combine the information to answer the question. Since John died in 2005, it is likely that the question is asking about the husband of Helen Robson Kemper Walton after John's death. Since Helen was married to Sam Walton, the correct answer is that there is no husband of Helen Robson Kemper Walton after John's death, as Sam Walton passed away before John.
Therefore, the answer is: There is no answer (as Helen Robson Kemper Walton was already married to Sam Walton and John had passed away).

——————————————
CoT–4: $P(r_4|X) = 0.125$
——————————————

Step 1: Identify the prominent Walton family member who died after John died in 2005. The prominent Walton family member who died after John died in 2005 is Helen Robson Kemper Walton.

Step 2: Determine who Helen's husband was. Helen's husband was Sam Walton, the founder of Walmart and Sam's Club.
Step 3: Since John died in 2005, it is likely that Helen's husband John is referring to John Walton, who died in 2005. This means Christy Walton took John's place after his death, as stated in the context.
Step 4: Therefore, the husband of the prominent Walton family member who died after John died in 2005 is John Walton.
Therefore, the answer is: John Walton.

——————————————
CoT–5: $P(r_5|X) = 0.125$
——————————————

Step 1: Identify the prominent Walton family member who died after John died in 2005. Helen Robson Kemper Walton died in 2007, after John Walton's death in 2005.
Step 2: Determine who Helen Robson Kemper Walton was married to. Helen Robson Kemper Walton was married to Sam Walton.
Step 3: Determine who took Sam Walton's place after his death. Sam Walton's wife Helen took her husband's place after his death.
Step 4: Identify who took John Walton's place after his death. Christy Walton took her husband John's place after his death.
Step 5: Combine the information to answer the question. Since Christy Walton took John's place after his death, it can be inferred that John was Christy's husband.
Step 6: Therefore, the answer is: John Walton.

——————————————
CoT–6: $P(r_6|X) = 0.025$
——————————————

Step 1: Identify the Walton family member who died after John died in 2005. The Walton family member who died after John died in 2005 is Helen Robson Kemper Walton.
Step 2: Determine who Helen Robson Kemper Walton was married to. According to the context, Helen was the wife of Sam Walton, the founder of Walmart and Sam's Club.

Step 3: Since John is mentioned as dying in 2005, it implies that Helen's husband is not John, but rather Sam Walton. However, the question asks about the husband of the prominent Walton family member who died after John died in 2005, which would be Helen.
Step 4: Since Helen is the only prominent Walton family member mentioned as dying after John died in 2005, it can be inferred that her husband is the one being asked about. However, the question is asking about the husband of the prominent Walton family member who died after John died in 2005, not Helen herself. This implies that the question is actually asking about the husband of Christy Walton, who took her husband John's place after his death.
Step 5: Therefore, the answer is: John Walton, who was Christy Walton's husband.

——————————————
CoT−7: $P(r_7|X) = 0.05$
——————————————
Step 1: Identify the Walton family member who died after John died in 2005. Helen Robson Kemper Walton died in 2007, after John's death in 2005.
Step 2: Determine who Helen's husband was. Helen's husband was Sam Walton, the co−founder of Walmart.
Step 3: However, the question asks for the husband of the prominent Walton family member who died after John died in 2005. Since Helen died in 2007, it cannot be her.
Step 4: The question does not provide the exact date of Christy Walton's husband John's death, but it mentions that Christy took her husband John's place after his death. Therefore, it can be inferred that Christy Walton's husband is the answer to the question.
Step 5: Combine the information to answer the question. The husband of the prominent Walton family member who died after John died in 2005 is John Walton's husband, which is Christy Walton's husband, also named John.
Therefore, the answer is: John.

——————————————
CoT−8: $P(r_8|X) = 0.025$
——————————————
Step 1: Identify the prominent Walton family member who died after John died in 2005. Helen Robson Kemper Walton died in 2007.
Step 2: Determine the relationship between Helen and John. Helen was the wife of John.
Step 3: Identify the question being asked. The question is asking for the husband of the prominent Walton family member who died after John died in 2005.
Step 4: Combine the information to answer the question. Since Helen died after John in 2005, John is the husband of the prominent Walton family member who died after John died in 2005.
Therefore, the answer is: John.

Intervention results

For the above eight chain−of−thoughts, we use the NWGM algorithm to perform causal intervention operations on them respectively. We can then compute the causal effect between the chain−of−thought and the answer.

$P(A = John|do(r_1)) = 1/10 = 0.1$
$P(A = SamWalton|do(r_1)) = 6/10 = 0.6$
$P(A = JohnWalton|do(r_1)) = 2/10 = 0.2$

$P(A = John|do(r_2)) = 3/10 = 0.3$
$P(A = SamWalton|do(r_2)) = 4/10 = 0.4$
$P(A = JohnWalton|do(r_2)) = 1/10 = 0.1$

$P(A = John|do(r_3)) = 4/10 = 0.4$
$P(A = SamWalton|do(r_3)) = 0/10 = 0.0$
$P(A = JohnWalton|do(r_3)) = 1/10 = 0.1$

$P(A = John|do(r_4)) = 0/10 = 0.0$
$P(A = SamWalton|do(r_4)) = 2/10 = 0.2$
$P(A = JohnWalton|do(r_4)) = 8/10 = 0.8$

$P(A = John|do(r_5)) = 0/10 = 0.0$
$P(A = SamWalton|do(r_5)) = 0/10 = 0.0$
$P(A = JohnWalton|do(r_5)) = 5/10 = 0.5$

$P(A = John|do(r_6)) = 0/10 = 0.0$
$P(A = SamWalton|do(r_6)) = 0/10 = 0.0$
$P(A = JohnWalton|do(r_6)) = 8/10 = 0.8$

$P(A = John|do(r_7)) = 8/10 = 0.8$
$P(A = SamWalton|do(r_7)) = 0/10 = 0.0$
$P(A = JohnWalton|do(r_7)) = 2/10 = 0.2$

$P(A = John|do(r_8)) = 6/10 = 0.6$
$P(A = SamWalton|do(r_8)) = 0/10 = 0.0$
$P(A = JohnWalton|do(r_8)) = 1/10 = 0.1$

Final results

The final answer is obtained by performing a weighted voting as follows:
$P(A = John|do(X))$ = 0.325 $*$ 0.1 + 0.25 $*$ 0.3 + 0.075 $*$ 0.4 + 0.125 $*$ 0.0 + 0.125 $*$ 0.0 + 0.025 $*$ 0.0 + 0.05 $*$ 0.8 + 0.025 $*$ 0.6 = 0.1925

$P(A = SamWalton|do(X))$ = 0.325 $*$ 0.6 + 0.25 $*$ 0.4 + 0.075 $*$ 0.0 + 0.125 $*$ 0.2 + 0.125 $*$ 0.0 + 0.025 $*$ 0.0 + 0.05 $*$ 0.0 + 0.025 $*$ 0.0 = 0.32

$P(A = JohnWalton|do(X))$ = 0.325 $*$ 0.2 + 0.25 $*$ 0.1 + 0.075 $*$ 0.1 + 0.125 $*$ 0.8 + 0.125 $*$ 0.5 + 0.025 $*$ 0.8 + 0.05 $*$ 0.2 + 0.025 $*$ 0.1 = 0.2925
Finally, we chose the answer with the largest weight as the final answer.
Therefore, the final answer obtained according to the Causal Prompting method is Sam Walton.

H Prompt Templates

In this section, we introduce the prompt templates of Chain-of-thought prompting (detailed in Section 3.1), CoT Improvement based on NWGM approximation (detailed in Section 3.2), Samples generation for Contrastive Learning (detailed in Section 3.4) and Demonstration Construction (detailed in Appendix C.2), respectively. The blue texts in prompts are required for LLM completion.

H.1 Chain-of-thought prompting

CoT Prompt template of Multi-hop Question Answering task.

Instruction

You are a helpful assistant to perform Multi–hop Question Answering. Based on the context, answer the question step by step and provide the final answer in the end.

Demonstration

Q:
The context is: [paragraphs]
The question is: [question]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

Test example:

Q:
The context is: [paragraphs]
The question is: [question]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

CoT Prompt template of GSM8K dataset.

Instruction

You are a helpful assistant to perform Mathematical reasoning. Answer the question step by step and provide the final answer in the end.

Demonstration

Q:
The question is: [question]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

Test example:

Q:
The question is: [question]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

CoT Prompt template of MATH dataset.

Instruction

You are a helpful assistant to perform Mathematical reasoning. Answer the question step by step and provide the final answer in the end. Presented in Latex format in text mode. Your answer should be inside \boxed{ }, such as \boxed{answer}.

Demonstration

Q:
The question is: [problem]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

Test example:

Q:
The question is: [problem]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

CoT Prompt template of Aspect-based Sentiment Analysis (ABSA) task.

Instruction

You are a helpful assistant to perform sentiment classification. Please detect the sentiment polarity towards the target given the sentence. The sentiment polarities include positive, negative and neutral. Please focus on sentiment of the target itself. Detect the sentiment polarity step by step and provide the final answer in the end.

Demonstration

Q:
The sentence is: [text]
The target is: [target].
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

Test example:

Q:
The sentence is: [text]
The target is: [target].
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

CoT Prompt template of Natural Language Inference (NLI) task.

Instruction

You are a helpful assistant to perform Natural language inference. Natural language inference is the task of determining whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given a "premise". Answer in a consistent style. Please write the reasoning process before giving the answer. Please provide your answer in the last sentence of your response. Your answer should be entailment, contradiction or neutral.

Demonstration

Q:
The premise is: [premise]
The hypothesis is: [hypothesis]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

Test example:

Q:
The premise is: [premise]
The hypothesis is: [hypothesis]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

CoT Prompt template of Fact Verification (FV) task.

Instruction

You are a helpful assistant to perform fact verification. Please check the veracity of the claim according to the evidence, including SUPPORTS and REFUTES. Answer in a consistent style. Please write the reasoning process before giving the answer. Please provide your answer in the last sentence of your response. Your answer should be either SUPPORTS or REFUTES.

Demonstration

Q:
The claim is: [question]
The evidence is: [context]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

Test example:

Q:
The claim is: [question]
The evidence is: [context]
Let us think step by step.
A:
Sure! Let us think step by step. [cot]
Therefore, the final answer is: [answer]

H.2 CoT Improvement based on NWGM approximation

We show the prompt templates of CoT Improvement based on NWGM approximation for the seven datasets listed below, including HotpotQA, MuSiQue, GSM8K, MATH, ABSA, NLI and FV. Among them, HotpotQA and MuSiQue datasets share the same template.

Prompt template of Multi-hop Question Answering task.

Instruction

You are a helpful assistant to perform Multi–hop Question Answering. Based on the context, answer the question step by step and provide the final answer in the end. I will provide a reasoning process, and please improve the reasoning process and make sure you get the correct answer. Give your final answer using the "The answer is:" format.

Demonstration

Q:
The context is: [paragraphs]
The question is: [question]
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [correct_cot]
Therefore, the correct answer is: [answer]

Test example:

Q:
The context is: [paragraphs]
The question is: [question]
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [improved_cot]
Therefore, the correct answer is: [answer]

Prompt template of GSM8K dataset.

Instruction

You are a helpful assistant to perform Mathematical reasoning. Answer the question step by step and provide the final answer in the end. I will provide a reasoning process, and please improve the reasoning process and make sure you get the correct answer.

Demonstration

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

Test example:

Q:
The question is: [question]
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [improved_cot]
Therefore, the correct answer is: [answer]

Prompt template of MATH dataset.

Instruction

You are a helpful assistant to perform Mathematical reasoning. Answer the question step by step and provide the final answer in the end. I will provide a reasoning process, and please improve the reasoning process and make sure you get the correct answer. Presented in Latex format in text mode. Your answer should be inside \boxed{}, such as \boxed{answer}.

Demonstration

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

Test example:

Q:
The question is: [problem]
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [improved_cot]
Therefore, the correct answer is: [answer]

Prompt template of Aspect-based Sentiment Analysis (ABSA) task.

Instruction

You are a helpful assistant to perform sentiment classification. Please detect the sentiment polarity towards the target given the sentence. The sentiment polarities include positive, negative and neutral. Please focus on sentiment of the target itself. Detect the sentiment polarity step by step and provide the final answer in the end. I will provide a reasoning process, and please improve the reasoning process and make sure you get the correct answer.

Demonstration

Q:
The sentence is: [text]
The target is: [target].
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [correct_cot]
Therefore, the correct answer is: [answer]

Test example:

Q:
The sentence is: [text]
The target is: [target].
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [improved_cot]
Therefore, the correct answer is: [answer]

Prompt template of Natural Language Inference (NLI) task.

Instruction

You are a helpful assistant to perform Natural language inference. Natural language inference is the task of determining whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given a "premise". Answer in a consistent style. Please write the reasoning process before giving the answer. Please provide your answer in the last sentence of your response. Your answer should be entailment, contradiction or neutral. I will provide a reasoning process, and please improve the reasoning process and make sure you get the correct answer.

Demonstration

Q:
The premise is: [premise]
The hypothesis is: [hypothesis]
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [correct_cot]
Therefore, the correct answer is: [answer]

Test example:

Q:
The premise is: [premise]
The hypothesis is: [hypothesis]
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [improved_cot]
Therefore, the correct answer is: [answer]

Prompt template of Fact Verification (FV) task.

Instruction

You are a helpful assistant to perform fact verification. Please check the veracity of the claim according to the evidence, including SUPPORTS and REFUTES. Answer in a consistent style. Please write the reasoning process before giving the answer. Please provide your answer in the last sentence of your response. Your answer should be either SUPPORTS or REFUTES. I will provide a reasoning process, and please improve the reasoning process and make sure you get the correct answer.

Demonstration

Q:
The claim is: [question]
The evidence is: [context]
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [correct_cot]
Therefore, the correct answer is: [answer]

Test example:

Q:
The claim is: [question]
The evidence is: [context]
Let us think step by step.
The provided reasoning process is: [wrong_cot]
A:
The improved reasoning process is: [improved_cot]
Therefore, the correct answer is: [answer]

H.3 Samples generation for Contrastive Learning

Prompt template of samples generation for Contrastive Learning.

Instruction

You are an expert in data augmentation. Please generate similar sentences based on the sentences I provided. Don't generate extraneous content.

Test example:

Q:
Provided sentences: [anchor_sentences]
A:
Positive sentences: [positive_sentences]

H.4 Demonstration Construction

Prompt template of demos generation for Multi-hop Question Answering task.

Instruction

You are a helpful assistant to perform Multi–hop Question Answering. Based on the context, answer the question step by step and provide the final answer in the end. I will provide the correct answer and ask you to write your thought process based on the answer.

Demonstration

Q:
The context is: [paragraphs]
The question is: [question]
The correct answer is: [answer]
Let us think step by step.
A:
The correct reasoning process is: [cot]

Test example:

Q:
The context is: [paragraphs]
The question is: [question]
The correct answer is: [answer]
Let us think step by step.
A:
The correct reasoning process is: [cot]

Prompt template of demos generation for Aspect-based Sentiment Analysis (ABSA) task.

Instruction

You are a helpful assistant to perform sentiment classification. Please detect the sentiment polarity towards the target given the sentence. The sentiment polarities include positive, negative and neutral. Please focus on sentiment of the target itself. Detect the sentiment polarity step by step and provide the final answer in the end. I will provide the correct answer and ask you to write your thought process based on the answer.

Demonstration

Q:
The sentence is: [text]
The target is: [target].
The correct answer is: [answer]
Let us think step by step.
A:
The correct reasoning process is: [cot]

Test example:

Q:
The sentence is: [text]
The target is: [target].
The correct answer is: [answer]
Let us think step by step.
A:
The correct reasoning process is: [cot]

Prompt template of demos generation for Natural Language Inference (NLI) task.

Instruction

You are a helpful assistant to perform Natural language inference. Natural language inference is the task of determining whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given a "premise". Answer in a consistent style. Please write the reasoning process before giving the answer. Please provide your answer in the last sentence of your response. Your answer should be entailment, contradiction or neutral. I will provide the correct answer and ask you to write your thought process based on the answer.

Demonstration

Q:
The premise is: [premise]
The hypothesis is: [hypothesis]
The correct answer is: [answer]
Let us think step by step.
A:
The correct reasoning process is: [cot]

Test example:

Q:
The premise is: [premise]
The hypothesis is: [hypothesis]
The correct answer is: [answer]
Let us think step by step.
A:
The correct reasoning process is: [cot]

Prompt template of demos generation for Fact Verification (FV) task.

Instruction

You are a helpful assistant to perform fact verification. Please check the veracity of the claim according to the evidence, including SUPPORTS and REFUTES. Answer in a consistent style. Please write the reasoning process before giving the answer. Please provide your answer in the last sentence of your response. Your answer should be either SUPPORTS or REFUTES. I will provide the correct answer and ask you to write your thought process based on the answer.

Demonstration

Q:
The claim is: [question]
The evidence is: [context]
The correct answer is: [answer]
Let us think step by step.
A:
The correct reasoning process is: [cot]

Test example:

Q:
The claim is: [question]
The evidence is: [context]
The correct answer is: [answer]
Let us think step by step.
A:
The correct reasoning process is: [cot]