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

Anonymous ACL submission

Abstract

Despite the significant achievements of existing prompting methods such as in-context learning and chain-of-thought for large language models (LLMs), they still face challenges of various biases. Traditional debiasing methods primarily focus on the model training stage, including data augmentation-based and reweight-based approaches, with the limitations of addressing the complex biases of 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 the bias of LLMs. In specific, causal intervention is implemented by designing the prompts without accessing the parameters and logits of LLMs.The chain-of-thoughts generated by LLMs are employed as the mediator variable and the causal effect between the input prompt and the output answers is calculated through front-door adjustment to mitigate model biases. Moreover, to obtain the representation of the samples precisely and estimate the causal effect more accurately, contrastive learning is used to fine-tune the encoder of the samples by aligning the space of the encoder with the LLM. Experimental results show that the proposed causal prompting approach achieves excellent performance on 3 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., 2022b), which allow LLMs perform natural language tasks based on only a few instances without weight updating. These prompting methods achieve significant results on 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).

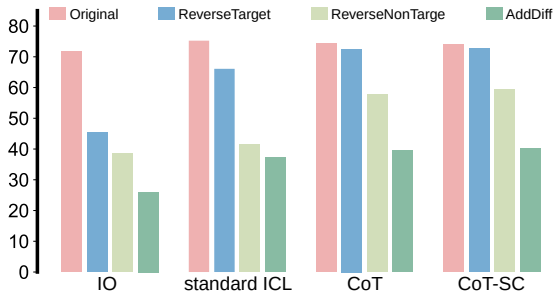

Figure 1: Performance of different prompting methods on ABSA (Pontiki et al., 2016) and its adversarial datasets. ReverseTarget, ReverseNonTarget, and AddDiff denote 3 different adversarial transformations by TextFlint. IO denotes the zero-shot setting where only the input question outputs the answer.

However, recent studies have shown that these prompting methods are not robust to some simple adversarial transformations (Ye et al., 2023). As shown in Figure 1, the performance of all prompting methods drops significantly 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 suffer from label bias, recency bias, and entity bias from context (Zhao et al., 2021; Wang et al., 2023a; Fei et al., 2023).

Traditional debiasing methods solve the bias problem mainly in the training stage of the model, including data augmentation-based (Wei and Zou, 2019; Lee et al., 2021) and reweight-based (Schuster et al., 2019; Mahabadi et al., 2019) methods. For data augmentation-based methods, it is costly to annotate and difficult to exhaust all bias cases due to the context length limitation. For reweight-based methods, it is impossible to assign weights to each sample in prompt-based learning scenarios. Recently, debias methods based on causal inference (Pearl et al., 2000; Pearl, 2022) have become

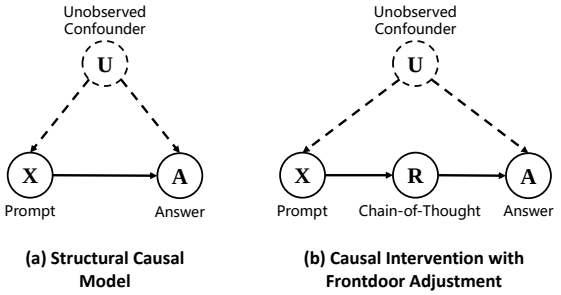

Figure 2: Structural causal model for the prompting method.

popular because of their strict theoretical guarantees and good generalization. Moreover, causal inference-based methods only need to calibrate the prediction results of the model in the inference stage (Niu et al., 2021; Tian et al., 2022; Guo et al., 2022; Xu et al., 2023; Chen et al., 2023), which is suitable for the prompt-based learning scenarios. However, counterfactual inference requires obtaining logits of LLM outputs while back-door adjustment requires modeling specific values of confounding variables.

Therefore, to address the above challenge, we propose to debias the prompting methods by causal intervention based on front-door adjustment (Pearl et al., 2016). Front-door adjustment allows causal intervention without access to confounding variable values and logits of LLM outputs. As shown in Figure 2(a), the causal relationship behind the prompting method is uncovered using a structural causal model. Here X denotes the input prompt, including demonstrations and test examples. A denotes the predicted answer generated by the LLM. U is the unobservable confounder that introduces various biases. The debiasing process measures 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 2(b), we use the chain-of-thoughts generated by LLMs as a mediator variable R between X and A. 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 designing the prompts without accessing the parameters and logits of LLMs. Specifically, to estimate the causal effect between X and R, we utilize self-consistency (Wang et al., 2022b) 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 best demonstration set, which can represent the expectation of the entire data distribution and help the model to generate an unbiased answer. Overall, CoT, self-consistency (SC), and ICL are effectively combined through front-door adjustment to mitigate the bias of LLMs on NLP tasks. Note that in the clustering and NWGM algorithms, an encoder is needed to obtain the representation of the samples. We use contrastive learning (Chen et al., 2020) to fine-tune the encoder to align the representation space of the encoder with the LLMs to estimate causal effects more accurately.

The contributions of this work are summarized as follows:

- To the best of our knowledge, our work is the first to uncover and analyze the bias problem in the prompting method of LLMs from the perspective of causal inference. Moreover, the front-door adjustment is proposed to solve the bias problem in prompting.
- Contrastive learning is proposed to fine-tune the encoder of the samples by aligning the space of the encoder with the LLM to obtain the representation of the samples precisely and estimate the causal effect more accurately.
- The proposed approach achieves excellent performance on 3 natural language processing datasets on both open-source and closed-source LLMs.

2 Related Work

2.1 Prompting Strategies

The performance of LLMs on downstream tasks largely depends on the prompting strategy. Adding a few labeled examples to the prompt can significantly improve the performance of LLM, sometimes even better than fine-tuned models (Brown et al., 2020; Chung et al., 2022; Dong et al., 2022). Following this way, recent work has proposed that including explanations and inference steps in the context of these examples can further improve

the quality of LLM responses (Nye et al., 2021; Lampinen et al., 2022; Wei et al., 2022). To improve the robustness of the results, some works sample from multiple answers generated by LLM based on the same prompt, and use majority voting to select the final answer (Wang et al., 2022b; Chen et al., 2021; Li et al., 2022).

Previous studies have shown that the performance of prompting methods is 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). In this paper, we mainly focus on example selection. Currently, the approaches for example selection aim to select the most relevant examples from the dataset. Rubin et al. (2021) and Liu et al. (2021) use the similarity of the sentence representation to select the most relevant samples. (Gonen et al., 2022) use the uncertainty of LLM to select the example with the lowest perplexity. Some works use BM25 (Wang et al., 2022a) and mutual information (Sorensen et al., 2022) for example selection. Recently, there have been other works based on active learning to select relevant examples (Diao et al., 2023; Margatina et al., 2023).

Different from these example selection approaches, we use the NWGM approximation to select the best demonstration set, which can represent the expectation of the entire dataset and help the model generate an unbiased answer.

2.2 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) and computer vision (Yang et al., 2021a).

Some works use counterfactual reasoning to remove the bias of the model (Xu et al., 2023; Guo et al., 2023; Niu et al., 2021).

Some recent work uses causal interventions for debiasing, including backdoor adjustment and front-door adjustment (Tian et al., 2022; Zhu et al., 2023; Wang et al., 2023a; Yang et al., 2021b).

Counterfactual inference requires obtaining logits of LLM outputs, and back-door adjustment requires modeling specific values of confounding variables, while front-door adjustment allows causal intervention without access to confounding variable values and logits of LLM outputs. Therefore, we propose to debias the prompting methods by causal intervention based on front-door adjustment. To the best of our knowledge, ours is the first work to apply front-door adjustment to the prompting method for debiasing.

3 Preliminaries

3.1 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 2(a), X denotes the input prompt, including demonstrations and test examples. A denotes the predicted answer generated by the LLM. LLM generates answers based on prompt, so we have $X \rightarrow A$ 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 fine-tuning datasets, leading to bias in downstream tasks. The reason is that the context of the pre-training data tends to follow a certain latent concept (Xie et al., 2021), and we use the unobservable variable U to describe this latent concept, using the back-door path $X \leftarrow U \rightarrow A$ denotes that the causality of X and A is confounded by U.

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 2(a), we should block $X \leftarrow U \rightarrow A$ 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 2(a), the do operation can be computed by back-door adjustment (Pearl et al., 2016):

$$P(A|do(X)) = \sum_{u} P(A|X, u)P(u) \quad (1)$$

3.2 Front-door Adjustment

Since we do not have access to the value of the confounding factor U, 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 2(b), we use the chain-of-thought generated by LLMs as a mediator variable R between X and A. Note that we focus on the confounder between X and A, so we decided to start with the simple SCM. Therefore, we ignore the confounder of R with other variables. According to the front door adjustment, we have

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

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

To compute $P(r|do(X))$, we need to block the backdoor path $X \leftarrow U \rightarrow A \leftarrow R$ between X and R. Since there exists a collision structure $U \rightarrow A \leftarrow R$, the backdoor path has been blocked (Pearl et al., 2016) and we have

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

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:

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

Finally, substituting Equations 3 and 4 into Equation 2, we obtain

$$\begin{aligned} P(A|do(X)) &= \sum_r P(r|do(X))P(A|do(r)) \\ &= \underbrace{\sum_r P(r|X)}_{CoT-SC} \underbrace{\sum_x P(x)P(A|r,x)}_{ICL} \end{aligned} \tag{5}$$

where the first half can be computed by combining the CoT and SC, and the second half can be computed by selecting the demonstration examples in ICL. More details are provided in Section 4.

4 Method

As shown in Figure 3, Causal Prompting aims to estimate the causal effect between input X and answer A, which can be divided into two parts with the front-door adjustment. First, the causal effect between X and reasoning chain r, $P(r|do(X))$ is estimated by combining the Chain-of-Thought prompting with a BERT-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 BERT-based normalized weighted geometric mean (NWGM) approximation algorithm. The final answer is aggregated by performing a weighted voting algorithm. Moreover, contrastive learning is employed to align the representation space of the BERT-based encoder and the LLMs for more precise estimation.

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

$P(r|do(X))$ measures the causal effect between input X and reasoning chain r. As shown in Equation 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 unavailable for most close-sourced LLMs, on the other hand, the reasoning chains r are difficult to be enumerated. Therefore, to estimate the causal effect $P(r|do(X))$ for both open-sourced and close-sourced LLMs, the CoT prompting and a BERT-based clustering algorithm are employed and combined. To be more specific, we first prompt the LLMs to generate multiple CoTs based on the input. Then, the CoTs are projected into text embeddings with a BERT-based encoder and clustered into clusters based on the embeddings. Finally, the center of each cluster is selected as the representative reasoning chain and the probability is estimated based on the cluster size.

For the input X, to improve the quality of generated CoTs, n in-context demonstrations d are created and concatenated with the question q^{test}.

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

It is worth noticing that the demonstrations d do not contain the final answer to prompt the LLMs to only generate the reasoning process and omit the final answer. Based on the input X, LLMs are prompted to generate m different CoTs c by increasing the temperature parameter of LLMs to encourage more random output, where the same procedure is also employed in self-consistency prompting of LLMs (Wang et al., 2023b).

$$\{c_i|i = 1, ..., m\} = \text{LLM}(X) \tag{7}$$

To perform the distance-based clustering method, the generated CoTs c are further fed into

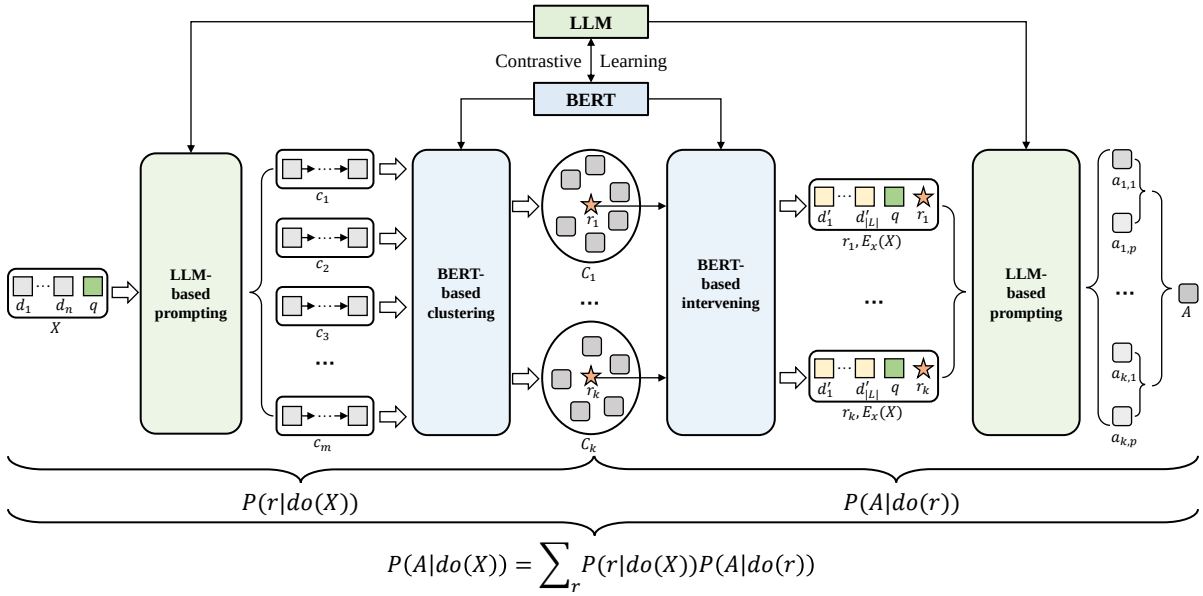

Figure 3: The framework of Causal Prompting.

a BERT-based encoder to get the text embedding $\boldsymbol{c}$. Following the previous work, the input is concatenated with the special tokens [CLS] and [SEP], and the embedding of the [CLS] token is taken as the embeddings of CoTs c.

$$\boldsymbol{c_i} = \text{BERT}([\text{CLS}], c_i, [\text{SEP}]) \tag{8}$$

Then K-means clustering is performed based on the embeddings to get k clusters C.

$$\{C_1, ..., C_k\} = \text{K-means}(\boldsymbol{c_1}, ..., \boldsymbol{c_m}) \tag{9}$$

Based on the clusters, k representative reasoning chains r are selected by taking the cluster centers.

$$r_i = \text{center}(C_i), i = 1, ..., k \tag{10}$$

and the probability is estimated based on the cluster size.

$$P(r_i|do(X)) \approx \frac{|C_i|}{m} \tag{11}$$

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

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

$P(A|do(r))$ measures the causal effect between the reasoning chain r and the answer A. Based on the discussion in Equation 4, $P(A|do(r))$ can be calculated with backdoor adjustment.

$$\begin{aligned} P(A|do(r)) &= \sum_x P(x)P(A|r,x) \\ &= \mathbb{E}_x[P(A|r,x)] \end{aligned} \tag{12}$$

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

However, the value space of x is inexhaustible in most of the cases, previous work employs the normalized weighted geometric mean (NWGM) approximation (Xu et al., 2015) to tackle this problem, where a confounder embedding is estimated to approximate the expectation of variable X.

$$\begin{aligned} \mathbb{E}_x[P(A|r,x)] &\approx P(A|r, \mathbb{E}_x[x]) \\ &\approx P(A|concat(r, \boldsymbol{x})) \end{aligned} \tag{13}$$

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

Inspired by the previous work, we proposed a prompt version of NWGM approximation to perform the back-door adjustment for LLMs prompting by combining a BERT-based intervention and In-Context Learning prompting. Following the previous works (Tian et al., 2022; Chen et al., 2023), we construct a confounding dictionary by coupling the input space and the label space. Then, the attention mechanism is employed to obtain the querying embeddings based on the confounding dictionary and reasoning chain. Finally, ICL demonstrations are selected by searching the entire training set based on the query embedding to approximate the effect of taking expectations on input space.

Assuming there are $|L|$ labels for the task performed (for the generation task, $|L|$ is set to 1), we construct a confounding dictionary Z by sampling

text embeddings of P demonstrations for each label.

$$\begin{aligned} \boldsymbol{z_j^{(l)}} &= \text{BERT}([\text{CLS}], z_j^{(l)}, [\text{SEP}]) \\ Z &= \{\boldsymbol{z_j^{(l)}} | l = 1, ..., |L|; j = 1, ..., P\} \end{aligned} \tag{14}$$

where $z_j^{(l)}$ denotes the reasoning chain of j-th demonstration for label l. The input space is approximated by the confounding dictionary Z.

Based on the confounding dictionary Z, the query embedding $\boldsymbol{q^{(l)}}$ for each label l is constructed for searching the training set for ICL demonstrations.

$$\begin{aligned} \boldsymbol{r} &= \text{BERT}([\text{CLS}], r, [\text{SEP}]) \\ \boldsymbol{q^{(l)}} &= \text{softmax}_j(\boldsymbol{z_j^{(l)}} \cdot \boldsymbol{r}) \boldsymbol{z_j^{(l)}} \end{aligned} \tag{15}$$

where $\boldsymbol{r}$ denotes the text embedding for reasoning chain r.

Then all instances in the training set are projected into text embeddings and the back-door intervention is approximated by searching the most similar instance based on query embeddings for each label l.

$$\begin{aligned} \boldsymbol{k_j^{(l)}} &= \text{BERT}([\text{CLS}], k_j^{(l)}, [\text{SEP}]) \\ d_l^{'} &= argmax_j(\boldsymbol{q^{(l)}} \cdot \boldsymbol{k_j^{(l)}}) \\ \mathbb{E}_x[x] &\approx [d_1^{'}, ..., d_{|L|}^{'}] \end{aligned} \tag{16}$$

where $k_j^{(l)}$ denotes the reasoning chain of j-th training instance for label l, $d_l^{'}$ denotes the selected demonstration for label l.

For each reasoning chain r_i, the final input after intervention consists of

$$X_{r_i}^{intervention} = [r_i, \mathbb{E}_x[x])] = [d_1^{'}, ..., d_{|L|}^{'}, q^{test}, r_i] \tag{17}$$

We prompt the LLMs p times to get p answers based on input $X_{r_i}^{intervention}$ for each r_i.

$$\{a_{i,j} | j = 1, ..., p\} = \text{LLM}(X_{r_i}^{intervention}) \tag{18}$$

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

$$P(A|do(r_i)) \approx \frac{\sum_{j=1}^{p} \mathbb{I}(A = a_{i,j})}{p} \tag{19}$$

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

Based on the results of Section 4.1 and Section 4.2, the final answer is obtained by performing a weighted voting.

$$\begin{aligned} P(A|do(X)) &= \sum_{r_i} P(r_i|do(X)) P(A|do(r_i)) \\ &= \sum_{i=1}^{k} \frac{|C_i|}{m} \cdot \frac{\sum_{j=1}^{p} \mathbb{I}(A = a_{i,j})}{p} \end{aligned} \tag{20}$$

Finally, we chose the answer with the largest weight as the final answer. See Algorithm 1 for the overall prompting process.

4.4 Representation Space Alignment

To align the representation spaces of the BERT encoder and the LLMs, we take each demonstration as an anchor, use LLM to generate the corresponding positive samples, and then use contrastive learning to finetune the BERT encoder.

For demonstration d_i, we prompt the LLM to generate a similar demonstration d_i^+ as the positive sample. Then we use the InfoNCE loss (Chen et al., 2020) to finetune the BERT encoder:

$$\sum_{\boldsymbol{d_p} \in Pos(i)} -log \frac{g(\boldsymbol{d_i}, \boldsymbol{d_p})}{g(\boldsymbol{d_i}, \boldsymbol{d_p}) + \sum_{j \in Neg(i)} g(\boldsymbol{d_i}, \boldsymbol{d_j})} \tag{21}$$

where the $\boldsymbol{d_i}$ and $\boldsymbol{d_p}$ are the representations of d_i and its positive samples. Pos and Neg refer to the positive set and the negative set for the demonstration d_i. $Pos(i) = \{\boldsymbol{d_{p1}}, \boldsymbol{d_{p2}}\}$, where $\boldsymbol{d_{p1}}$ is augmented representation of the same demonstration d_i, obtained with different dropout masks, and $\boldsymbol{d_{p2}}$ is the representation of d_i^+. $j \in Neg(i)$ is the index of in-batch negative samples. g is a function: $g(\boldsymbol{d_i}, \boldsymbol{d_j}) = exp(\boldsymbol{d_i^T} \boldsymbol{d_j}/t)$, where t is a positive value of temperature.

5 Experiments

5.1 Datasets

We evaluate the effectiveness of our approach on three tasks: Aspect-based Sentiment Analysis (ABSA), Natural Language Inference (NLI), and Fact Verification(FV). For the ABSA and NLI tasks, we use SemEval2014-Laptop (Pontiki et al., 2016) and MNLI-m (Williams et al., 2017) as the original (in-distribution, ID) datasets and the corresponding transformation data generated by TextFlint (Wang et al., 2021) as the adversarial (out-of-distribution, OOD) datasets. For the FV task, we use FEVER (Thorne et al., 2018) as the ID dataset and its adversarial dataset Symmetric

methods	ABSA			NLI			FV		
	ori #638	adv #1239	overall #1877	ori #819	adv #754	overall #1573	ori #239	adv #717	overall #956
standard ICL	**75.24**	48.51	57.59	56.29	37.67	47.36	72.8	65.97	67.68
CoT	74.29	54.0	60.9	52.99	30.9	42.4	83.26	68.34	72.07
CoT-SC	74.14	54.56	61.21	**59.22**	32.63	46.47	86.61	74.62	77.62
Causal Prompting	72.1	**60.13**	**64.2**	56.29	**41.51**	**49.21**	**93.31**	**78.8**	**82.43**

Table 1: Results on LLaMA2. **ori** denotes the original dataset (in-distribution) and **adv** denotes the adversarial dataset (out-of-distribution). The number after # indicates the number of samples. The best results are in bold.

methods	ABSA			NLI			FV		
	ori #638	adv #1239	overall #1877	ori #819	adv #754	overall #1573	ori #239	adv #717	overall #956
standard ICL	81.03	71.27	74.59	64.22	37.8	51.56	95.82	67.22	74.37
CoT	80.09	72.72	75.23	69.84	62.07	66.12	95.4	81.59	85.04
CoT-SC	78.68	73.77	75.44	**79.61**	62.6	71.46	**97.91**	81.73	85.77
Causal Prompting	**81.66**	**73.85**	**76.51**	79.37	**64.19**	**72.09**	**97.91**	**84.1**	**87.55**

Table 2: Results on GPT-3.5. The best results are in bold.

FEVER (Schuster et al., 2019) as the OOD dataset. Following previous work (Ye et al., 2023), for all datasets, we adopt the label classification accuracy as the evaluation metric.

5.2 LMs

We evaluate our prompting method on two Large Language Models: LLaMA2-7b-chat-hf (Touvron et al., 2023) in Transformers library (Wolf et al., 2019) and GPT-3.5-turbo-0125 (OpenAI, 2022).

We use BERT-base (Devlin et al., 2018) as the encoder in computing sentence similarity, clustering algorithm, and NWGM algorithm.

5.3 Baselines

We compare our approach with 3 other few-shot prompting approaches, including the **standard ICL** (Brown et al., 2020): Prompt LLMs with some demonstration examples containing only questions and answers; **CoT** (Wei et al., 2022): Demonstration examples include additional reasoning chains; **CoT-SC** (Wang et al., 2022b): We prompt LLM generates multiple different reasoning chains and use majority voting to select the final answer.

5.4 Settings

Demonstration Construction We use a few manually constructed demonstrations to prompt the LLM to generate reasoning chains and answers for all examples in the dataset, and keep the examples with correct answers to form the demonstration set. When selecting demonstrations, exclude demonstrations that are the same as the test examples. 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 For all prompting methods in this paper, we use sentence similarity to select the most relevant examples. For the classification task, to keep the label space balanced, it is guaranteed that there is one demonstration example for each category, that is, the number of demonstration examples in the prompt n is equal to the number of categories L for the corresponding NLP task. ABSA and NLI are 3-way classification tasks, and FV is a 2-way classification task.

Implementation Details The LLM parameters are the same for all prompting methods: temperature is set to 0.7, and top_p is set to 0.9. The number of votes in COT-SC is 50. In our method, the number of reasoning chains generated in the first part is $m = 50$, and then these reasoning chains are clustered into $k = 10$. For each reasoning chain representing the cluster center, we generated $p = 5$ answers based on the prompt modified by intervention. Finally, the $k \cdot p = 50$ answers were weighted voting to get the final answer. In the confounder dictionary used in the NWGM algorithm, the number of samples per category is $P = 10$.

5.5 Results

Tables 1 and 2 show the performance comparison results of Causal Prompting and other prompting methods on LLaMA2 and GPT-3.5, respectively. Note that for fair comparison, both Causal Prompting and CoT-SC perform majority voting on the same number of answers, as detailed in Section 5.4.

It can be observed that the **adv** and **overall** performance of Causal Prompting is the highest on all datasets for both LLaMA2 and GPT-3.5. This shows that our method generalizes well for both synthetic adversarial data (ABSA, NLI) generated by TextFlint and human-annotated real adversarial data (FV). The performance on LLaMA2 and GPT-3.5 demonstrates that Causal Prompting is effective for Large Language Models of different scales.

In Table 1, Causal Prompting performs worse than other methods on **ori** data for ABSA and NLI. In Table 2, the performance is lower than the other methods on **ori** data of NLI. Although the performance of Causal Prompting decreases on some **ori** data, the improvement is larger on **adv** data, resulting in the highest **overall** performance. In fact, Causal Prompting balances the performance of in-distribution and out-of-distribution data to improve the performance of the overall data. This phenomenon has also been reported in previous work on causal inference (Tian et al., 2022; Wang et al., 2023a).

5.6 Ablation Study

To explore the importance of the Contrastive Learning and Clustering algorithm for our causal intervention method, we separately evaluated the performance of the causal intervention after removing these two parts. w/o Contrastive Learning means that contrastive learning is no longer used to align the representation spaces of the encoder and the Large Language Models. w/o Cluster indicates that the Clustering algorithm is no longer used to estimate the probability of the causal intervention. Specifically, we use the $1/m$ estimate $P(r|do(X))$. Since the construction of confounding dictionary Z depends on the Clustering algorithm, we use random sampling demonstrations instead of the NWGM approximation in Equation 17.

As shown in Figure 4, the performance of causal prompting decreases after removing Contrastive Learning or Cluster, and the decrease is larger after removing Cluster. This indicates that Cluster has an important role in causal intervention.

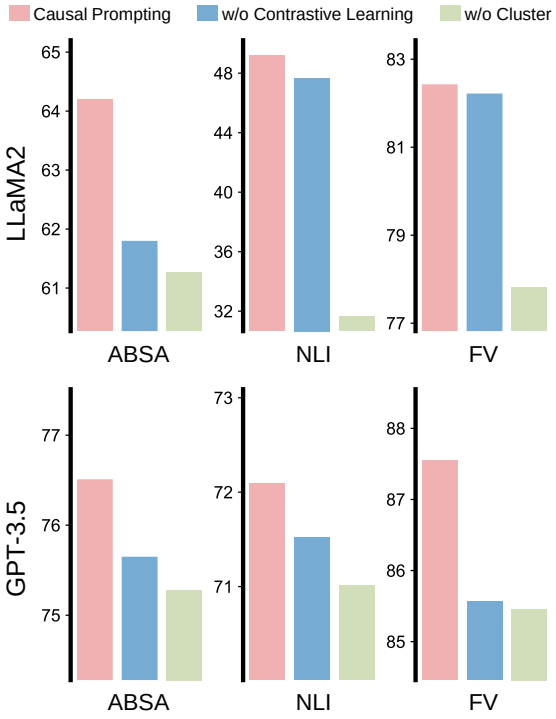

Figure 4: Ablation results.

In fact, w/o Cluster means a rough front-door adjustment, and the experimental results in Figure 4 show that the clustering algorithm can estimate the probability in the causal intervention more accurately, indicating that clustering can enhance the effect of causal intervention.

6 Conclusion

In this paper, we propose Causal Prompting, a prompting method based on front-door adjustment to effectively mitigate the bias of LLMs on NLP tasks. The chain-of-thought 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 parts, the causal effect between prompt and CoTs and the causal effect between CoTs and answer. The former part is estimated by combining the Chain-of-Thought prompting with a BERT-based clustering algorithm. The latter part is estimated by combining the In-Context-Learning prompting with the BERT-based 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. Experimental results show that Causal Prompting achieves excellent performance on 3 natural language processing datasets on both open-source and closed-source LLMs.

Limitations

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

- The three datasets in this paper are classification datasets, and we need to test the effectiveness of Causal Prompting on more datasets, such as open-domain question answering and mathematical reasoning datasets.
- As mentioned in Section 5.4, both Causal Prompting and CoT-SC are majority voting on the same number of answers $k \cdot p$, but Causal Prompting has two parts, so the total number generated by LLM is $m+k \cdot p$. Its computation is m more times than CoT-SC.
- We only evaluated the effectiveness of Causal Prompting on two Large Language Models, LLaMA2-7b and GPT-3.5, and we need to evaluate our method on more Large Language Models of different kinds and scales.

References

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

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. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

Ziwei Chen, Linmei Hu, Weixin Li, Yingxia Shao, and Liqiang Nie. 2023. 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.

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

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

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.

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

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. 2022. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158.

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

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*.

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

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

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

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

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

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

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

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

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

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

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

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

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. 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.

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

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

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

Judea Pearl. 2022. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea Pearl*, pages 373–392.

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

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

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

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. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)*, pages 19–30. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

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

Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*.

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

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

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

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

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022a. Training data is more valuable than you think: A simple and effective method by retrieving from training data. *arXiv preprint arXiv:2203.08773*.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, et al. 2021. 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.

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

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

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

Jason Wei and Kai Zou. 2019. 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)*.

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

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

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

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

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

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

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

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

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

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

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

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

A Causal Prompting Algorithm

As shown in Algorithm 1, we describe the operation flow of Causal Prompting. Where q^{test} is the test example. d is the demonstration examples set. n is the number of demonstration examples in the prompt. m is the number of reasoning chains generated in the first part of our method. k is the number of clusters in the first part of Causal Prompting. p is the number of answers generated based on the prompt modified by intervention in the second part of our method. Z is the confounding dictionary of the NWGM algorithm.

B Details of contrastive learning

For each LLM as well as for each task, we fine-tune one encoder using contrastive learning separately. During training, we set the batch size is 128. The learning rate is $1e-4$. The temperature t is set

Algorithm 1 Causal Prompting

Input: $d, q^{test}, n, m, p, k, Z$, BERT, LLM

$X \leftarrow [d_1, ..., d_n, q^{test}]$
$\{c_i | i = 1, ..., m\} \leftarrow \text{LLM}(X)$
$\boldsymbol{c_i} \leftarrow \text{BERT}([\text{CLS}], c_i, [\text{SEP}])$
$\{C_1, ..., C_k\} \leftarrow \text{K-means}(\boldsymbol{c_1}, ..., \boldsymbol{c_m})$
for $i = 1$ **to** k:
    $r_i \leftarrow \text{center}(C_i)$
    $P(r_i|X) \approx \frac{|C_i|}{m}$
end for
for $i = 1$ **to** k:
    $\boldsymbol{r_i} \leftarrow \text{BERT}([\text{CLS}], r_i, [\text{SEP}])$
    $\boldsymbol{q^{(l)}} \leftarrow \text{softmax}_j(\boldsymbol{z_j^{(l)}} \cdot \boldsymbol{r_i}) \boldsymbol{z_j^{(l)}}$
    $\boldsymbol{k_j^{(l)}} \leftarrow \text{BERT}([\text{CLS}], k_j^{(l)}, [\text{SEP}])$
    $d_l^{'} \leftarrow \text{argmax}_j(\boldsymbol{q^{(l)}} \cdot \boldsymbol{k_j^{(l)}})$
    $X_{r_i}^{intervention} \leftarrow [d_1^{'}, ..., d_{|L|}^{'}, q^{test}, r_i]$
    $\{a_{i,j} | j = 1, ..., p\} \leftarrow \text{LLM}(X_{r_i}^{intervention})$
    $P(A|do(r_i)) \approx \frac{\sum_{j=1}^{p} \mathbb{I}(A=a_{i,j})}{p}$
end for
$P(A|do(X)) \leftarrow \sum_{i=1}^{k} \frac{|C_i|}{m} \cdot \frac{\sum_{j=1}^{p} \mathbb{I}(A=a_{i,j})}{p}$
return $argmax_A(P(A|do(X)))$

to 0.3. The max length of BERT is 512. The total epochs is 50.

C More details on the adversarial datasets

Table 3 presents the descriptions of how multiple adversarial datasets are generated for ABSA, NLI, and FV tasks.

D Full experimental results

Table 4, 5, 6, 7, 8, 9 shows the more detailed performance comparison between Causal Prompting and other prompting methods on different adversarial categories dataset on LLaMA2 and GPT-3.5.

Table 10, 11, 12, 13, 14, 15 shows the more detailed performance results about ablation study on different adversarial categories dataset on LLaMA2 and GPT-3.5.

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

methods	ReverseTarget #466		ReverseNonTarget #135		AddDiff #638		Overall #1877
	ori	adv	ori	adv	ori	adv	
standard ICL	83.48	66.09	86.67	41.48	**75.24**	37.15	57.59
CoT	**83.91**	72.53	90.37	57.78	74.29	39.66	60.9
CoT-SC	**83.91**	**72.75**	89.63	59.26	74.14	40.28	61.21
Causal Prompting	83.05	72.53	**91.11**	**65.19**	72.1	**50.0**	**64.2**

Table 4: Results for ABSA task on LLaMA2. The best results are in bold.

methods	AddSent #417		NumWord #225		SwapAnt #333		Overall #1573
	ori	adv	ori	adv	ori	adv	
standard ICL	56.35	43.45	58.22	20.18	57.96	50.42	47.36
CoT	52.04	28.4	52.0	25.56	56.76	49.58	42.4
CoT-SC	**58.27**	29.85	**61.33**	26.91	**61.56**	**52.94**	46.47
Causal Prompting	54.44	**46.84**	54.67	**28.25**	60.66	47.9	**49.21**

Table 5: Results for NLI task on LLaMA2. The best results are in bold.

methods	Original #239	Symmetric #717	Overall #956
standard ICL	72.8	65.97	67.68
CoT	83.26	68.34	72.07
CoT-SC	86.61	74.62	77.62
Causal Prompting	**93.31**	**78.8**	**82.43**

Table 6: Results for FV task on LLaMA2. The best results are in bold.

methods	ReverseTarget #466		ReverseNonTarget #135		AddDiff #638		Overall #1877
	ori	adv	ori	adv	ori	adv	
standard ICL	84.55	75.54	**93.33**	55.56	81.03	71.47	74.59
CoT	86.05	75.75	92.59	72.59	80.09	70.53	75.23
CoT-SC	86.05	**77.47**	91.85	72.59	78.68	71.32	75.44
Causal Prompting	**86.48**	75.75	**93.33**	**75.56**	**81.66**	**72.1**	**76.51**

Table 7: Results for ABSA task on GPT-3.5. The best results are in bold.

methods	AddSent #417		NumWord #225		SwapAnt #333		Overall #1573
	ori	adv	ori	adv	ori	adv	
standard ICL	64.51	44.42	64.44	13.45	66.97	60.5	51.56
CoT	71.46	67.96	72.89	**47.98**	71.77	**68.07**	66.12
CoT-SC	78.18	78.4	80.0	38.57	**83.78**	52.94	71.46
Causal Prompting	**78.9**	**80.58**	**80.44**	37.22	82.88	57.98	**72.09**

Table 8: Results for NLI task on GPT-3.5. The best results are in bold.

methods	Original #239	Symmetric #717	Overall #956
standard ICL	95.82	67.22	74.37
CoT	95.4	81.59	85.04
CoT-SC	**97.91**	81.73	85.77
Causal Prompting	**97.91**	**84.1**	**87.55**

Table 9: Results for FV task on GPT-3.5. The best results are in bold.

methods	ReverseTarget #466		ReverseNonTarget #135		AddDiff #638		Overall #1877
	ori	adv	ori	adv	ori	adv	
Causal Prompting	83.05	72.53	**91.11**	**65.19**	72.1	**50.0**	**64.2**
w/o Contrastive Learning	83.26	**72.75**	89.63	60.74	73.51	42.32	61.8
w/o Cluster	**83.48**	72.1	90.37	58.52	**73.82**	41.38	61.27

Table 10: Ablation study results for ABSA task on LLaMA2.

methods	AddSent #417		NumWord #225		SwapAnt #333		Overall #1573
	ori	adv	ori	adv	ori	adv	
Causal Prompting	**54.44**	**46.84**	54.67	**28.25**	**60.66**	47.9	**49.21**
w/o Contrastive Learning	53.0	45.63	**55.56**	24.66	59.16	**51.26**	47.68
w/o Cluster	35.25	25.24	41.33	18.39	40.84	35.29	31.66

Table 11: Ablation study results for NLI task on LLaMA2.

methods	Original #239	Symmetric #717	Overall #956
Causal Prompting	**93.31**	**78.8**	**82.43**
w/o Contrastive Learning	**93.31**	78.52	82.22
w/o Cluster	91.21	73.36	77.82

Table 12: Ablation study results for FV task on LLaMA2.

methods	ReverseTarget #466		ReverseNonTarget #135		AddDiff #638		Overall #1877
	ori	adv	ori	adv	ori	adv	
Causal Prompting	**86.48**	75.75	**93.33**	**75.56**	**81.66**	**72.1**	**76.51**
w/o Contrastive Learning	85.62	77.68	91.85	74.07	78.37	71.79	75.65
w/o Cluster	85.84	**77.9**	91.85	73.33	79.0	70.06	75.28

Table 13: Ablation study results for ABSA task on GPT-3.5.

methods	AddSent #417		NumWord #225		SwapAnt #333		Overall #1573
	ori	adv	ori	adv	ori	adv	
Causal Prompting	**78.9**	**80.58**	**80.44**	37.22	82.88	**57.98**	**72.09**
w/o Contrastive Learning	77.94	77.67	79.56	**40.36**	**85.59**	52.1	71.52
w/o Cluster	77.94	79.37	80.0	35.87	83.18	54.62	71.01

Table 14: Ablation study results for NLI task on GPT-3.5.

methods	Original #239	Symmetric #717	Overall #956
Causal Prompting	**97.91**	**84.1**	**87.55**
w/o Contrastive Learning	97.07	81.73	85.56
w/o Cluster	**97.91**	81.31	85.46

Table 15: Ablation study results for FV task on GPT-3.5.