

# Large Language Models Are Better Logical Fallacy Reasoners with Counterargument, Explanation, and Goal-Aware Prompt Formulation

Jiwon Jeong<sup>1</sup> Hyeju Jang<sup>2\*</sup> Hogun Park<sup>1\*</sup>

<sup>1</sup>Sungkyunkwan University, Republic of Korea

<sup>2</sup>Indiana University Indianapolis, USA

jwjw9603@skku.edu hyejuj@iu.edu hogunpark@skku.edu

## Abstract

The advancement of Large Language Models (LLMs) has greatly improved our ability to process complex language. However, accurately detecting logical fallacies remains a significant challenge. This study presents a novel and effective prompt formulation approach for logical fallacy detection, applicable in both supervised (fine-tuned) and unsupervised (zero-shot) settings. Our method enriches input text incorporating implicit contextual information—*counterarguments*, *explanations*, and *goals*—which we query for validity within the context of the argument. We then rank these queries based on confidence scores to inform classification. We evaluate our approach across multiple datasets from 5 domains, covering 29 distinct fallacy types, using models from the GPT and LLaMA series. The results show substantial improvements over state-of-the-art models, with F1 score increases of up to 0.60 in zero-shot settings and up to 0.45 in fine-tuned models. Extensive analyses further illustrate why and how our method excels.

## 1 Introduction

Logical fallacies are flawed arguments resulting from faulty reasoning (Clark, 1971). For example, consider the statement: “Annie must like Starbucks because all girls like Starbucks.” This exemplifies the logical fallacy known as *faulty generalization*, as it assumes that all girls universally share a preference for Starbucks—an overly broad generalization. Recognizing logical fallacies is essential for both humans and AI. It enables us to avoid being misled by faulty arguments, improves communication by minimizing errors, prevents misleading or deceptive misinformation, and strengthens convincing arguments (Woods, 2004; Tindale, 2007).

Despite this importance, logical fallacy detection remains in its early stage in natural language

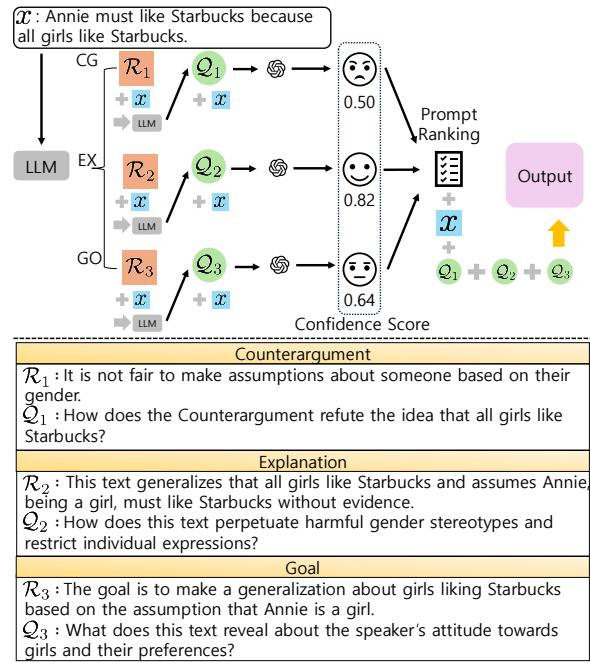


Figure 1: Prompt formulation:  $x$  represents the input text to classify.  $R_i$  denotes the contextual augmentation generated from the input text using specific instructions for Counterargument (CG), Explanation (EX), and Goal (GO).  $Q_i$  denotes the reformulated queries created from each augmentation to analyze the input text.

processing. Recent large language models (LLMs) have demonstrated challenges not only in general reasoning (Naveed et al., 2023; Chen et al., 2024) but also in detecting logical fallacies. For example, Jin et al. (2022) showed that both zero-shot and fine-tuning results of BERT-based models are suboptimal. More recently, Hong et al. (2024) highlighted that most LLMs continue to struggle in this area. To the best of our knowledge, there has been no attempt to address this issue through prompting engineering or other unsupervised methods.

To address this gap, we investigate whether incorporating implicit information from multiple perspectives on arguments into prompts can improve LLM performance in logical fallacy detection. We

\*Corresponding authors.

propose a novel two-step prompt formulation process that leverages implicit contextual information (See Figure 1), applicable in both supervised (fine-tuned) and unsupervised (zero-shot) settings. Inspired by Sourati et al. (2023), we leverage three key types of implicit information relevant to arguments: *Counterargument*, *Explanation*, and *Goal*. We hypothesize that this implicit information provides additional context for identifying logical fallacies.

To operationalize this idea, we augment input text in the prompt with this implicit contextual information. Specifically, we begin by generating statements summarizing the relevant background (counterargument, explanation, and goal), followed by queries that assess the validity of this information within the argument’s context. These queries are then ranked using a prompt ranking method based on confidence scores. Then, this ranking information is integrated into the input prompts. Notably, all steps of this process are automatically generated, requiring no manual prompt adjustments.

We evaluate our approach on five publicly available datasets, covering 29 distinct fallacy types, using models from the GPT and LLaMA series. Our experiments show that this approach substantially outperforms baselines and state-of-the-art models in both unsupervised and supervised settings. Specifically, compared to state-of-the-art models, we achieve up to an improvement of 0.60 in Macro-F1 score in zero-shot settings and 0.45 in fine-tuned models. Although Sourati et al. (2023) pioneered the use of this type of information, how we extract and use the information differs substantially, resulting in superior performance. We also perform extensive analyses to explore why and how our method excels, including examining the impact of our prompt ranking method, performing calibration analysis, and conducting error analysis. Our code is available at [https://github.com/jw9603/Logical\\_Fallacy](https://github.com/jw9603/Logical_Fallacy).

Our contributions can be summarized in three ways. (1) We introduce a structured prompt design for logical fallacy detection that effectively integrates three types of implicit contextual information. This goes beyond previous methods by using a confidence-based prompt ranking system, which incorporates diverse information to enhance the model’s analysis. (2) We perform extensive evaluation across five datasets, demonstrating the effectiveness of our approach across various domains. (3) We conduct extensive in-depth analyses

to validate our methodology, examining the effectiveness of each component in design.

## 2 Related Work

### 2.1 Logical Fallacy Detection

Previous research explored computational methods for logical fallacy detection in various contexts, including dialogues (Habernal et al., 2017), argument sufficiency (Stab and Gurevych, 2017; Wachsmuth et al., 2017), Reddit discussions (Sahai et al., 2021), misinformation (Musi et al., 2022), and educational materials (Jin et al., 2022). These studies focused on single datasets, limiting their ability to demonstrate generalizability across diverse domains with varying argument structures, which is crucial for robust fallacy detection in real-world scenarios.

Most recent approaches aimed to address this issue by tackling multiple datasets. For instance, Alhindhi et al. (2022) enhanced fallacy detection performance across multiple datasets by employing a T5 model (Raffel et al., 2020) with multi-task instruction-based prompting, outperforming single-dataset models. Sourati et al. (2023) proposed a Case-Based Reasoning method, enriching cases with implicit information, including *counterarguments*, *goals*, *explanations*, and *structure*, to retrieve similar cases. While both approaches demonstrated effectiveness, they rely on supervised methods which require large datasets.

Furthermore, Hong et al. (2024) evaluated various LLMs using prompting techniques. They experimented with providing definitions of logical fallacies within sentences and without, but found little improvement in performance. This study revealed that LLMs exhibit limited performance in fallacy detection, emphasizing the need for specialized reasoning methods tailored to the task.

Similar to prior research, our study aims to detect logical fallacies across multiple datasets, but it is differentiated by its focus on leveraging LLMs more effectively in both supervised and unsupervised settings. We build on the idea of using implicit contextual information, *counterarguments*, *goals*, and *explanations*, introduced by Sourati et al. (2023). While our results corroborate their findings on the value of such information for the task, our work diverges significantly from theirs by focusing on prompt engineering rather than case-based retrieval. Although they demonstrated the utility of the implicit information for logical fallacy detection, their work does not explore how LLMs can

generate or utilize this information. In contrast, our research systematically integrates these contextual elements into LLMs’ reasoning process, thereby improving classification accuracy without requiring extensive labeled data. This shift is crucial, as it moves the focus from manual case-based reasoning to an automated, scalable solution leveraging LLMs’ in-context learning capabilities.

## 2.2 Prompting Method for Contextual Augmentation

There has been similar work using prompts for contextual augmentation and query reformulation. Shen et al. (2024) refine queries using retrieved results to improve search performance. Yang et al. (2023) propose Question-Driven Visual Exploration (QVix), generating exploratory questions to enhance Large Vision-Language Models (LVLMs)’ ability to analyze visual content. While both approaches leverage LLMs to enhance query formulation or reasoning, they focus on retrieval optimization or visual understanding, respectively, rather than logical analysis. Unlike these prior approaches, our method constructs structured reasoning prompts specifically designed to facilitate logical fallacy classification, ensuring that LLMs engage in systematic argument evaluation.

Additionally, it is worth noting that our use of the term “query” differs from its role in information retrieval. In our context, “query” represents a question designed to guide logical fallacy classification, rather than a search input for document retrieval.

## 3 Our Approach

Our approach consists of four main steps to generate structured prompts. As shown in Figure 1, we first use the LLM to create contextual augmentations, and based on these augmentations and the input text, we generate context-informed queries. Next, we classify logical fallacies using the generated queries, and finally, we rank the queries based on their confidence scores, incorporating this ranking information into the final classification.

We focus on three types of implicit information related to arguments for augmenting input text: *Counterargument*, *Explanation*, and *Goal*. This contextual information provides diverse insights into logical fallacies, as demonstrated by Sourati et al. (2023). Counterarguments present alternative viewpoints (Nussbaum et al., 2005), explanations dissect the logic and reasoning (Hempel and Op-

Data	N	C	Genre	Domain
PROPAGANDA	12,267	16†	News	Politics
ARGOTARIO	1,338	6†	Dialogue	General
LOGIC	2,449	13	Dialogue	Education
COVID-19	154	11†	SocMed/News	Covid-19
CLIMATE	685	11†	News	Climate

Table 1: Summary of five fallacy datasets. N: # of samples, C: # of classes. †: No-Fallacy class included.

penheim, 1948), and goals evaluate whether the argument supports its conclusion (Tracy, 2013).

### Step 1: Generate Contextual Augmentation

First, we use LLMs to generate three types of implicit information, i.e., *Counterargument*, *Explanation*, and *Goal*. Each type of information is denoted by an index  $i$ , representing the specific instruction applied ( $\mathcal{I}^1$  for Counterargument,  $\mathcal{I}^2$  for Explanation, and  $\mathcal{I}^3$  for Goal). These instructions guide the LLM to generate contextual augmentations  $\mathcal{R}_i$ . By applying specific instructions  $\mathcal{I}^i$  to gpt-3.5-turbo-instruct, we generate a contextual augmentation:

$$\mathcal{R}_i = \text{LLM}(x, \mathcal{I}^i). \quad (1)$$

For instance, for the statement  $x = \text{“Annie must like Starbucks because all girls like Starbucks.”}$ , the Goal perspective prompt is “*Express the goal of the text*,” leading to  $\mathcal{R}_i = \text{“The goal of this text is to make a generalization about girls liking Starbucks based on the assumption that Annie is a girl.”}$  This contextually augmented text is used for generating a query.

### Step 2: Generate Reformulated Queries

To generate context-informed queries, we design a tailored query generation method using the LLM instruction: “*Create one query for each text to analyze the text based on its goal rather than directly asking what a logical fallacy is.*” This process yields  $\mathcal{Q}_i = \text{“What does this text reveal about the speaker’s attitude towards girls and their preferences?”}$  Such queries enable a nuanced understanding of the underlying assumptions and biases in arguments:

$$\mathcal{Q}_i = \text{LLM}(x, \mathcal{R}_i). \quad (2)$$

### Step 3: Calculate Confidence Scores for Queries

In this step, we calculate the confidence scores for predicting the probability  $p_{\text{LLM}}$  of the fallacy’s class label ( $l$ ) with each contextual augmentation  $i$  for an input text  $x$ :

$$\text{conf}(\mathcal{Q}_i) = \max_{l \in L} \sum_{\text{tokens}} \log p_{\text{LLM}}(l|x, \mathcal{Q}_i),$$

where  $L$  denotes a set of logical fallacy labels. Here,  $\text{conf}(\mathcal{Q}_i)$  represents the highest confidence score derived from the log probabilities of the predicted tokens, indicating how confidently the model predicts the most probable class based on the specific query type. The reformulated queries and their respective confidence scores will later be used in the following step to finalize the classification through ranking-based methods.

**Step 4: Prompt Ranking-Based Classification**  
To predict a final label considering all contexts, we use the confidence scores calculated in the previous step. Specifically, the confidence scores  $\text{conf}(\mathcal{Q}_i)$  for each query are sorted in descending order and incorporated into the final prompt as ranking information as follows:

$$\hat{y} = \arg \max_{l \in L} p_{\text{LLM}}(l|x, Q_{\text{all}}, \text{Rank}(Q_{\text{all}})),$$

where  $Q_{\text{all}} = \{\mathcal{Q}_i, \forall i\}$ . Here,  $\mathcal{Q}_i$  contains the actual content of each query (e.g., Counterargument, Explanation, Goal). The function,  $\text{Rank}(Q_{\text{all}}) = \text{Sorted}(\{\text{conf}(\mathcal{Q}_i), \forall i\})$ , ranks the queries based on their respective confidence scores. This ranking is then converted into text and included as part of the prompt. Both the content and the ranking information are utilized for enhancing classification performance. Further details on the prompting methods and examples can be found in Appendix B.2.

## 4 Evaluation

To evaluate our method, we conduct experiments on multi-class classification to identify specific types of logical fallacies.

### 4.1 Datasets

We experiment with the five fallacy datasets: PROPAGANDA (Da San Martino et al., 2019), ARGOTARIO (Habernal et al., 2017), LOGIC (Jin et al., 2022), COVID-19 (Musi et al., 2022), and CLIMATE (Alhindi et al., 2022).

These datasets cover 5 different domains and 29 unique fallacy types, as shown in Table 1. This breadth allows us to assess the generalizability of our method.

### 4.2 Baselines

We compare our method against several baselines in both unsupervised and supervised settings. In the unsupervised setting, we first use a baseline zero-shot approach that detects logical fallacies in a given text without using any additional information. We also compare our model against previous work, including Zero-shot CoT (ZCoT) (Kojima et al., 2022), which utilizes a Chain of Thought approach, and DEF (Hong et al., 2024), which provides the model with fallacy definitions. Further details on ZCoT and DEF are in Appendix B.1.

In the supervised setting, we compare our method against state-of-the-art models proposed by Jin et al. (2022), Sourati et al. (2023), and Alhindi et al. (2022). Sourati et al. (2023) employ contextual augmentations within a case-based reasoning framework, while Alhindi et al. (2022) apply multi-task learning for fallacy detection.

Jin et al. (2022) employs a structure-aware classifier that leverages additional coreference annotations to learn structural patterns in arguments, which is only applicable to the LOGIC dataset. We report the results from the original papers for (Jin et al., 2022) and (Alhindi et al., 2022), and run the code provided by Sourati et al. (2023) to obtain their results.

### 4.3 Model Implementation Details

For the unsupervised setting, we use LLMs from the GPT and LLaMA series with different sizes: gpt-3.5-turbo<sup>1</sup>, gpt-4<sup>2</sup>, Llama2-7b-hf, Llama2-13b-hf, and Llama3-8B.

For the supervised setting, we implement our approach by fine-tuning a roberta-base model (Liu, 2019) for fallacy classification. We concatenate the input text with each query as input to the model. The prompt ranking process is excluded in this setting, as it relies on confidence scores from LLMs like gpt-3.5-turbo. We train the roberta-base model using the AdamW optimizer (Loshchilov and Hutter, 2019) on two NVIDIA RTX A6000 GPUs, which takes ~30 minutes. The batch size is chosen from {4, 8, 16, 32, 64} and the learning rate from {1e-5, 2e-5}. These hyperparameters are tuned using the development set.

<sup>1</sup>We use gpt-3.5-turbo-0125.

<sup>2</sup>We use gpt-4-0613.

LM	Supervised	Method	PROPAGANDA		ARGOTARIO		LOGIC		COVID-19		CLIMATE	
			ACC	F1								
Fine-tuned LMs	✓	(Jin et al., 2022)	—	—	—	—	0.48	0.59	—	—	—	—
		(Sourati et al., 2023)	<b>0.95</b>	<b>0.91</b>	0.71	0.71	0.76	0.71	0.36	0.18	0.48	0.29
		(Alhindi et al., 2022)	0.73	0.56	0.64	0.64	0.70	0.66	0.29	0.28	0.25	0.20
		roberta-base + EX (Ours)	0.91	0.84	<b>0.81</b>	<b>0.80</b>	<b>0.80</b>	<b>0.79</b>	<b>0.36</b>	0.17	<b>0.84</b>	<b>0.74</b>
gpt-3.5-turbo	✗	Zero-shot	0.12	0.07	0.50	0.41	0.39	0.25	0.29	0.14	0.18	0.13
		ZCoT (Kojima et al., 2022)	0.14	0.08	0.48	0.39	0.39	0.23	0.29	0.11	0.20	0.13
		DEF (Hong et al., 2024)	0.23	0.11	0.50	0.48	0.40	0.22	0.29	0.11	0.16	0.09
		Prompt Ranking (Ours)	<b>0.35</b>	<b>0.17</b>	<b>0.84</b>	<b>0.69</b>	<b>0.45</b>	<b>0.32</b>	<b>0.71</b>	<b>0.56</b>	<b>0.63</b>	<b>0.58</b>
gpt-4	✗	Zero-shot	0.32	0.19	0.60	0.50	0.40	0.28	0.36	0.25	0.18	0.11
		ZCoT (Kojima et al., 2022)	0.31	<b>0.20</b>	0.56	0.47	0.41	0.28	0.43	0.33	0.17	0.12
		DEF (Hong et al., 2024)	0.34	0.17	0.60	0.52	0.40	0.28	0.43	0.29	0.18	0.14
		Prompt Ranking (Ours)	<b>0.40</b>	<b>0.20</b>	<b>0.84</b>	<b>0.83</b>	<b>0.49</b>	<b>0.37</b>	<b>0.86</b>	<b>0.76</b>	<b>0.78</b>	<b>0.71</b>
Llama3-8B	✗	Zero-shot	0.13	0.05	0.21	0.13	0.25	0.13	0.07	0.02	0.14	0.08
		ZCoT (Kojima et al., 2022)	0.23	0.04	0.19	0.13	0.20	0.09	0.07	0.07	0.13	0.06
		DEF (Hong et al., 2024)	<b>0.28</b>	0.06	0.25	0.13	0.23	0.11	0.07	0.03	0.14	0.07
		Prompt Ranking (Ours)	0.22	<b>0.09</b>	<b>0.30</b>	<b>0.29</b>	<b>0.38</b>	<b>0.36</b>	<b>0.36</b>	<b>0.25</b>	<b>0.33</b>	<b>0.40</b>

Table 2: Multi-class fallacy classification performance. **Bold**: the highest score for each base model. **Bold**: the highest score across all methods for each dataset in the unsupervised setting. Results for (Jin et al., 2022) and (Alhindi et al., 2022) are taken from their original papers, with the same test splits used for comparison. —: The model requires additional annotations to run. roberta-base + EX (Ours) refers to our supervised approach using Explanation (EX) queries, which show the best performance among our proposed query types.

#### 4.4 Results

**Overall Performance** Table 2 shows the accuracy and Macro-F1 scores for multi-class fallacy classification across all datasets. The table includes the best-performing LLaMA model for the unsupervised setting in addition to two GPT models, and the best-performing supervised model (roberta-base + EX) among our three queries. For comprehensive results, refer to Appendix D and Figure 2.

Our prompt ranking method overall outperforms other approaches in unsupervised settings. For instance, gpt-4 with prompt ranking achieves the highest scores on the PROPAGANDA dataset (accuracy: 0.40, Macro-F1: 0.20). In the supervised settings, roberta-base with Explanation queries shows the best performance on the CLIMATE dataset, achieving an accuracy of 0.84 and a Macro-F1 score of 0.74.

**Performance per Query Type** Figure 2 shows the overall performance of each query type. A consistent trend is observed across datasets and models: Explanation (EX) queries yield the best overall performance, excelling across diverse datasets and achieving highest scores in both accuracy and F1 metrics. Goal (GO) queries also perform well, particularly in datasets like PROPAGANDA, where the content often has a clear objective or hidden intent, making the GO queries effective at identifying the underlying purpose behind persuasive arguments. However, Counterargument (CG) queries

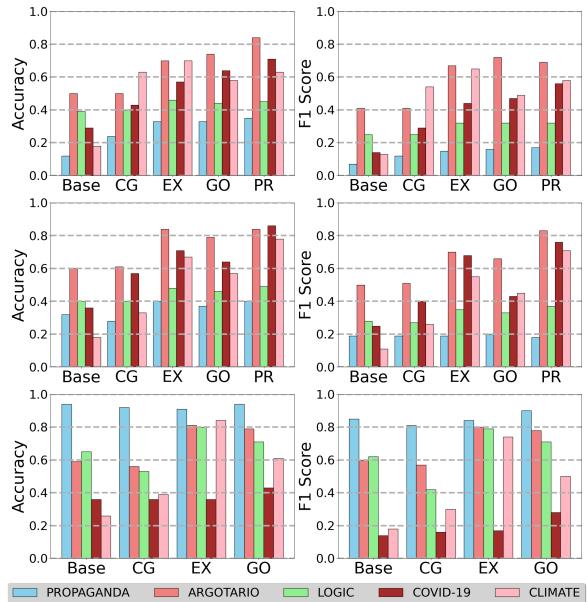


Figure 2: Multi-class classification results based on query types for all datasets using gpt-3.5-turbo, gpt-4, and roberta-base from top to bottom. CG: Counterargument, EX: Explanation, GO: Goal, and PR: Prompt Ranking. Base: the method that uses only logical fallacy sentences without any queries.

consistently underperform relative to EX and GO queries. While CG queries contribute to detection performance, their overall effectiveness is lower, particularly when compared to the stronger results achieved with EX and GO queries.

Figure 3 shows the rankings of each query type across 29 fallacy classes. EX queries consistently

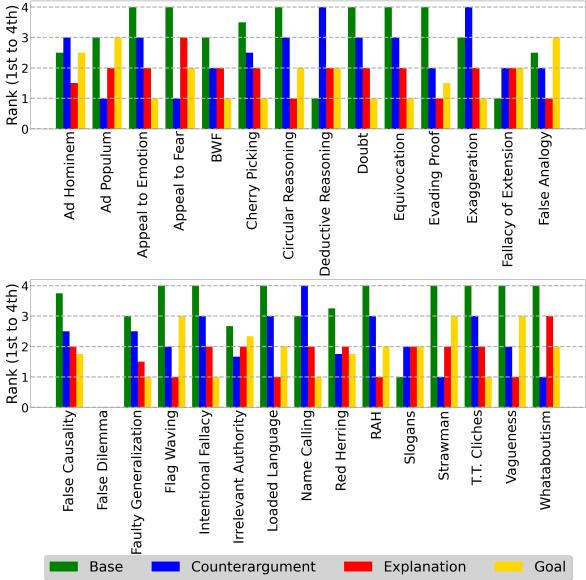


Figure 3: Performance comparison of base (without queries) and all query types across various fallacy classes using the gpt-3.5-turbo model. The y-axis represents the average rank of each method across datasets. Lower ranks indicate better performance. BWF: Black and White Fallacy, RAH: Reductio Ad Hitlerum, T.T. Cliches: Thought Terminating Clichés.

perform best, followed by GO queries, with CG queries ranking lowest. EX queries excel in fallacies requiring detailed reasoning, like False Analogy and Evading the Burden of Proof, while GO queries perform well in intent-driven fallacies, such as Appeal to Emotion and Intentional Fallacy. CG queries, though generally less effective, are suited for simpler logical fallacies like Ad Populum and Strawman.

Some exceptions stand out. In the Slogans fallacy, the Base method outperforms all query types, possibly because its simplicity is more effective for direct and emotionally charged statements. Similarly, in the Flag Waving fallacy, EX queries outperform GO, suggesting that identifying logical inconsistencies may sometimes be more effective than focusing on emotional intent. Overall, the queries consistently outperform the Base method, demonstrating their value in fallacy detection.

## 5 Further Analyses

### 5.1 Performance Based on Confidence Scores

We further investigate how the model’s performance with each individual query type varies based on different confidence scores, which refer to the probability assigned by the LLM when generating tokens in the query, prior to the prompt ranking step.

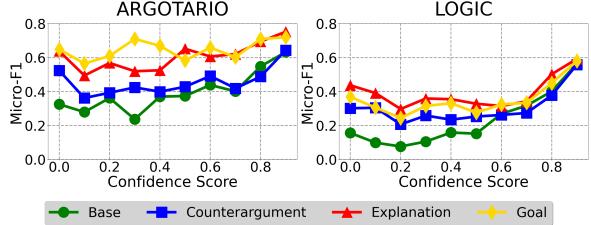


Figure 4: Relationship between confidence scores and performance with/without queries for two datasets using the gpt-3.5-turbo model.

We measure Micro-F1 scores to assess overall performance across all classes, rather than focusing on class-level performance, using the gpt-3.5-turbo model. Figure 4 illustrates the results for the ARGOTARIO and LOGIC datasets. The performance trends are consistent across other datasets.

The results indicate that incorporating queries consistently improves F1 scores across different confidence scores in all datasets. In particular, at lower confidence scores, the performance gap between our model and the baseline is more pronounced. This suggests that queries provide valuable contextual information, helping to clarify the underlying logic of the text.

### 5.2 Calibration Analysis for Prompt Ranking

We aim to show that the Prompt Ranking (PR) method achieves better calibration compared to the Base method, which does not use queries, by evaluating both methods using the gpt-3.5-turbo model. Calibration is assessed using the Reliability Diagram (Guo et al., 2017), which compares predicted confidence with observed accuracy, revealing deviations from perfect calibration, such as overconfidence or underconfidence. The calibration results for the ARGOTARIO and LOGIC datasets are presented in Figure 5.

Overall, while our PR demonstrates better calibration compared to the Base method, the LLMs still show some degree of overconfidence, consistent with findings from previous studies (Yin et al., 2023; Ren et al., 2025). This trend is observed across other datasets such as PROPAGANDA, COVID-19, and CLIMATE, where PR consistently outperforms the Base method, further underscoring its effectiveness in leveraging multiple context-informed queries to provide a more nuanced interpretation of the input text and reduce confidence biases. Since the final prediction is made by ranking the three confidence scores, our

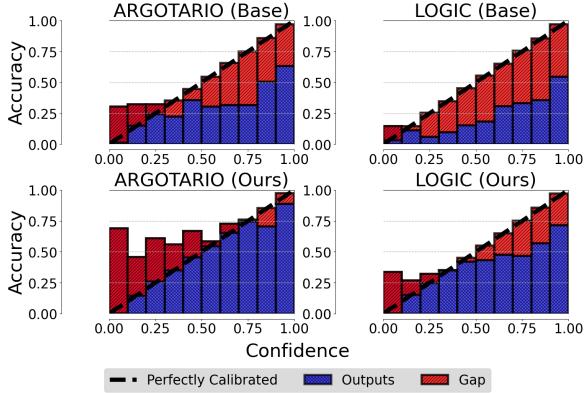


Figure 5: Reliability diagrams comparing the calibration of the base method (without queries) and ours (prompt ranking) using the gpt-3.5-turbo model across two datasets.

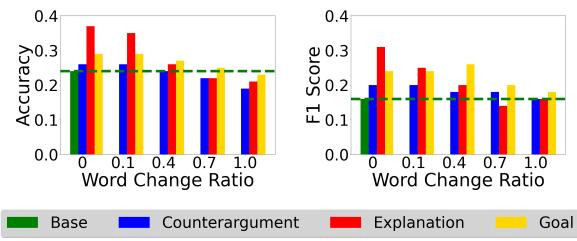


Figure 6: Impact of word change ratio on accuracy and Macro-F1 score for each query type.

model demonstrates high accuracy even when confidence scores are low.

### 5.3 Impact of Ranking Information

To investigate the factors behind the success of our prompt ranking method, we conduct experiments to determine whether providing the ranking information is essential for performance gains. In the None setting, all three queries—Counterargument, Explanation, and Goal—are presented without any ranking, meaning the ranking information is not provided to the model. In the Random setting, the queries are randomly shuffled, disregarding their confidence scores, and this randomized ranking information is provided.

Table 3 shows that Prompt Ranking consistently outperforms both Random and None settings, underscoring the importance of confidence-based ranking in improving performance. For example, in the CLIMATE dataset, gpt-4 with prompt ranking achieves an accuracy of 0.78 and an F1 score of 0.71, while Random Ranking results in a significant drop in the F1 score, highlighting the value of explicitly including the ranking information.

### 5.4 Impact of Query Structure

This analysis aims to determine whether the performance gains of our model in fallacy detection come from the specific structure of our queries (i.e., counterargument, explanation, and goal) or simply from adding extra information, regardless of accuracy or relevance. To test this, we compare our structured queries to randomly generated ones. We hypothesize that if our queries significantly outperform random ones, it would confirm that their structure is key to the observed performance gains. By replacing words in the reformulated queries with arbitrarily similar words, we assess the impact on logical reasoning.

For the analysis, we randomly select 100 samples from the validation sets of five datasets. To maximize diversity, we repeat the randomization process five times and choose the set with the highest number of unique fallacy classes, resulting in 100 samples representing 18 distinct fallacy types.

To generate random queries, we replace content words in the structured queries with semantically related words obtained from ConceptNet (Speer et al., 2017), by selecting the nearest neighbors of each target word. This approach generates queries that are topically relevant but lack the counterargument, explanation, and goal structure. We opt for topically relevant replacements rather than entirely random words to ensure a fair comparison. Although other resources like WordNet (Miller, 1995) could be used, ConceptNet is chosen for its broader range of semantic relationships, though we believe the choice of resource would not significantly affect the analysis. Figure 6 shows that performance declines as the word change rate increases. This demonstrates that the effect of our method relies on structured logical reasoning, not merely the addition of some information.

### 5.5 Impact of Our Approach for Leveraging Implicit Information

To investigate the effect of our approach for leveraging implicit information, we compare it to the method used by Sourati et al. (2023), which also incorporates implicit information. The details of the previous method are provided in Appendix B.1. We generate implicit information using both our prompts and theirs and compare the performance of using the information generated by each. We use the gpt-3.5-turbo-instruct model to generate contextual augmentations and queries.

LLM	Method	PROPAGANDA		ARGOTARIO		LOGIC		COVID-19		CLIMATE	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
gpt-3.5-turbo	None	0.33	0.16	0.71	0.67	0.44	0.30	0.64	0.56	0.61	0.53
	Random	0.33 $\pm$ 0.01	<b>0.17</b> $\pm$ 0.01	0.76 $\pm$ 0.02	0.63 $\pm$ 0.06	0.41 $\pm$ 0.01	0.29 $\pm$ 0.01	0.58 $\pm$ 0.05	0.43 $\pm$ 0.06	0.31 $\pm$ 0.03	0.25 $\pm$ 0.03
	Prompt Ranking (Ours)	<b>0.35</b>	<b>0.17</b>	<b>0.84</b>	<b>0.69</b>	<b>0.45</b>	<b>0.32</b>	<b>0.71</b>	<b>0.56</b>	<b>0.63</b>	<b>0.58</b>
gpt-4	None	0.39	0.19	0.81	0.80	0.48	0.36	0.93	0.85	0.72	0.59
	Random	0.40 $\pm$ 0.01	0.19 $\pm$ 0.02	0.83 $\pm$ 0.01	0.77 $\pm$ 0.06	0.48 $\pm$ 0.01	0.36 $\pm$ 0.01	0.79 $\pm$ 0.03	0.71 $\pm$ 0.06	0.31 $\pm$ 0.03	0.25 $\pm$ 0.01
	Prompt Ranking (Ours)	<b>0.40</b>	<b>0.20</b>	<b>0.84</b>	<b>0.83</b>	<b>0.49</b>	<b>0.37</b>	<b>0.86</b>	<b>0.76</b>	<b>0.78</b>	<b>0.71</b>

Table 3: Impact of ranking information from our prompt ranking method across five datasets. None: not providing ranking information explicitly, Random: providing the randomized order of the three queries as ranking information, and results for this method are averaged over five runs (seeds 0 to 4), ACC: accuracy, F1: Macro-F1 score, **Bold**: the highest score for each model, **Bold**: the highest score across all models for each dataset.

Method	Acc	P	R	F1
<i>Base</i>	0.12	0.20	0.13	0.07
Contextual Augmentations ( <a href="#">Sourati et al., 2023</a> )				
+ Counterargument	0.13	0.12	0.11	0.08
+ Explanation	0.23	0.14	0.18	0.11
+ Goal	0.19	0.16	0.17	0.11
Reformulated Query based on <a href="#">Sourati et al. (2023)</a>				
+ Counterargument	0.12	0.11	0.13	0.07
+ Explanation	0.21	0.15	0.14	0.09
+ Goal	0.16	0.13	0.12	0.08
Contextual Augmentations (Ours)				
+ Counterargument	0.19	0.15	0.12	0.11
+ Explanation	0.20	0.13	0.15	0.10
+ Goal	0.28	0.16	0.18	0.14
Reformulated Query from (Ours)				
+ Counterargument	0.24	0.19	0.16	0.12
+ Explanation	<b>0.33</b>	0.17	0.19	0.15
+ Goal	<b>0.33</b>	<b>0.20</b>	<b>0.20</b>	<b>0.16</b>

Table 4: Performance comparison based on different contextual augmentation prompt methods and their resulting queries on the PROPAGANDA dataset using the gpt-3.5-turbo model. Acc: Accuracy, P: precision, R: recall, and F1: Macro-F1 score.

As shown in Table 4, our queries outperform theirs on the PROPAGANDA dataset. For example, when using our queries based on Explanation and Goal, the classification accuracy is 0.33, compared to 0.23 when using their method. The same trend is observed across other datasets, confirming that our approach for generating contextual augmentations and queries consistently outperforms theirs in detecting logical fallacies.

## 5.6 Illustrative Examples of Query Impact

Table 5 provides examples illustrating how different query types (Counterargument (CG), Explanation (EX), Goal (GO)) affect the model’s performance compared to the Base method, which does not use any queries.

The first example illustrates a situation where the Base method fails to detect the fallacy, but all three query types successfully guide the model to

the correct classification. Specifically, CG focuses on evaluating judgments about the speaker’s work ethic, EX examines how personal attacks shift attention away from the main argument, and GO highlights how these personal attacks detract from the discussion of television effectiveness on children’s minds. This shows how queries can help the model focus on the relevant reasoning behind the argument and accurately identify the fallacy.

In contrast, the next two examples highlight cases where the Base method correctly identifies logical fallacies but fails when queries are applied. In these cases, both EX and GO queries only confirm the presence of a fallacy without challenging the logic behind it, while the CG queries introduce unrelated topics, such as societal stereotypes or regional differences, further distracting the model from identifying the correct fallacy.

These findings suggest that queries can hinder and improve model performance, depending on how well queries align with the intricate logical structure of the argument. When queries fail to direct the model to analyze the core logic or introduce extraneous information, as seen with CG queries in the second and third examples—they can impair performance. Conversely, when queries focus the model on critical aspects of the argument, as in the first example, they can improve performance. Therefore, the effectiveness of each query type depends on its alignment with the underlying logic of the argument, which merits further exploration to determine whether this can be achieved through zero-shot prompting in future work.

## 6 Conclusion

In this paper, we presented a simple yet powerful prompt formulation approach for logical fallacy detection. We demonstrated the value of incorporating additional contextual information structured as counterarguments, explanations, and goals, com-

Example	Prediction
[Base] Question: Is television an effective tool in building the minds of children?, Answer: You want to sit around and watch tv? Better get a job.	Red Herring (✗)
[CG] How do judgments about a person’s work ethic affect the argument about television being an effective tool for developing children’s minds?	Ad Hominem (✓)
[EX] Does the speaker’s personal attack distract from the important question being posed and avoid addressing it?	Ad Hominem (✓)
[GO] How does the use of personal attacks in this response detract from the discussion about the effectiveness of television on children’s minds?	Ad Hominem (✓)
[Base] Why would you lend money to some who looks like they don’t have a job?	Ad Hominem (✓)
[CG] How can we combat the harmful and inaccurate societal stereotype that someone’s appearance is a reliable indicator of their financial situation?	Faulty Generalization (✗)
[EX] What is the flaw in the argument presented in this statement?	Faulty Generalization (✗)
[GO] Can you identify any logical fallacies in this text and explain how they affect the overall argument?	Faulty Generalization (✗)
[Base] There’s no trend in global droughts. Cold snaps in the U.S. are down but, unexpectedly, so are heatwaves.	False Causality (✓)
[CG] How might regional differences or potential data manipulation impact the conclusion that there is no trend in global droughts?	Cherry Picking (✗)
[EX] What is the logical fallacy in this text and why is it problematic?	Cherry Picking (✗)
[GO] What is the logical fallacy present in this text and what is the author’s goal in pointing it out?	Cherry Picking (✗)

Table 5: Illustrative examples of query impact on gpt-4 performance. Base: the method without using any queries, CG: Counterargument, EX: Explanation, GO: Goal.

bined with prompt ranking based on confidence scores. Our extensive experiments consistently showed improvements over baselines and state-of-the-art models in both unsupervised and supervised settings, highlighting the robustness and adaptability of our approach.

## 7 Limitations

While this study demonstrates promising advancements in leveraging Large Language Models (LLMs) like GPT-4 for detecting and classifying logical fallacies through a novel prompt formulation approach, we acknowledge several limitations.

First, the five datasets used in our evaluation may not fully capture the diversity and complexity of real-world scenarios, such as logical reasoning in fields like medicine and science. Additionally, there are fallacies beyond the 29 types we explored. For instance, the type-token fallacy, which occurs when a word can refer to either a type (an abstract descriptive concept) or a token (an object that instantiates a concept) and is used ambiguously, was not included in our analysis.

Furthermore, our method’s reliance on specific models (gpt-3.5-turbo, gpt-4, Llama2-7b-hf, Llama2-13b-hf, and Llama3-8B) may limit its applicability to other LLMs or future iterations, potentially impacting scalability and cost-effectiveness due to the computational resources required for

prompt reformulation.

In addition, while the improvements in Macro-F1 scores are significant, further research is needed to enhance the interpretability and transparency of the reformulation process to ensure broader applicability in real-world settings.

Lastly, variations in query performance across fallacy classes may stem from differences in dataset-specific definitions and class imbalance. For instance, in the datasets for our experiments, more prevalent classes like Loaded Language allowed for clearer evaluations, while underrepresented classes like Strawman limited meaningful comparisons. This underscores the need to address class imbalance to ensure more consistent evaluations in future studies.

## 8 Ethics Statement

Our methodology aims to enhance logical fallacy detection by employing contextual augmentations and structured prompt reformulation. We do not utilize any external knowledge or information that might bias the evaluation of our model. However, while our approach improves model performance in identifying logical fallacies, it may inadvertently amplify existing biases present in the training data of LLMs. We acknowledge this limitation and recommend further research to mitigate potential biases in logical reasoning tasks.

## 9 Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & evaluation (IITP) grant, and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2019-II190421, IITP-2025-RS-2020-II201821, RS-2024-00438686, RS-2024-00436936, RS-2024-00360227, RS-2023-0022544, NRF-2021M3H4A1A02056037, RS-2024-00448809). This research was also partially supported by the Culture, Sports, and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024 (RS-2024-00333068). Further support was provided by Indiana University Indianapolis.

## References

- Tariq Alhindi, Tuhin Chakrabarty, Elena Musi, and Smaranda Muresan. 2022. Multitask instruction-based prompting for fallacy recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 8172–8187.
- Xinyun Chen, Ryan A Chi, Xuezhi Wang, and Denny Zhou. 2024. Premise order matters in reasoning with large language models. In *Proceedings of the International Conference on Machine Learning*.
- Michael Clark. 1971. Review of c.l. hamblin, fallacies. *Philosophical Books*, 12.
- Giovanni Da San Martino, Yu Seunghak, Alberto Barrón-Cedeno, Rostislav Petrov, Preslav Nakov, et al. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 5636–5646.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330.
- Ivan Habernal, Raffael Hannemann, Christian Polak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational argumentation meets serious games. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 7–12.
- Carl G Hempel and Paul Oppenheim. 1948. Studies in the logic of explanation. *Philosophy of Science*, 15(2):135–175.
- Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. 2024. A closer look at the self-verification abilities of large language models in logical reasoning. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.
- Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyou Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schoelkopf. 2022. Logical fallacy detection. In *Proceedings of the Findings of the Association for Computational Linguistics*, pages 7180–7198.
- 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.
- Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Elena Musi, Myrto Aloumpi, Elinor Carmi, Simeon Yates, and Kay O'Halloran. 2022. Developing fake news immunity: fallacies as misinformation triggers during the pandemic. *Online Journal of Communication and Media Technologies*, 12(3).
- Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.
- E Michael Nussbaum, CarolAnne M Kardash, and Steve Ed Graham. 2005. The effects of goal instructions and text on the generation of counterarguments during writing. *Journal of Educational Psychology*, 97(2):157.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2025. Investigating the factual knowledge boundary of large language models with retrieval augmentation. In *Proceedings of the International Conference on Computational Linguistics*.

Saumya Sahai, Oana Balalau, and Roxana Horincar. 2021. Breaking down the invisible wall of informal fallacies in online discussions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 644–657.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Dixin Jiang. 2024. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *Findings of the Association for Computational Linguistics*, pages 15933–15946.

Zhivar Sourati, Filip Ilievski, Hông-ÂN Sandlin, and Alain Mermoud. 2023. Case-based reasoning with language models for classification of logical fallacies. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 5188–5196.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Christian Stab and Iryna Gurevych. 2017. Recognizing insufficiently supported arguments in argumentative essays. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 980–990.

Christopher W Tindale. 2007. *Fallacies and argument appraisal*. Cambridge University Press.

Karen Tracy. 2013. *Understanding face-to-face interaction: Issues linking goals and discourse*. Routledge.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017. Computational argumentation quality assessment in natural language. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 176–187.

John Hayden Woods. 2004. *The death of argument: Fallacies in agent based reasoning*, volume 32. Springer Science & Business Media.

Kaiwen Yang, Tao Shen, Xinmei Tian, Xiubo Geng, Chongyang Tao, Dacheng Tao, and Tianyi Zhou. 2023. Good questions help zero-shot image reasoning. *arXiv preprint arXiv:2312.01598*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuan-Jing Huang. 2023. Do large language models know what they don’t know? In *Findings of the Association for Computational Linguistics*, pages 8653–8665.

## A Overview

In this appendix, we provide detailed explanations and additional results related to the methods used in the main study. Section B introduces the prompt techniques, including ZCoT, DEF, and the contextual augmentation method, along with examples of our four-step prompt technique. Section C outlines the grouping of similar fallacy classes across datasets, as well as the dataset splits for supervised and unsupervised settings. Section D presents multi-class classification results for LLaMA models. Finally, Section E provides additional information on calibration results across five datasets for Base, Ours (Prompt Ranking), ZCoT (Kojima et al., 2022), DEF (Hong et al., 2024), and individual queries.

## B Prompt Details

### B.1 Prompt Techniques: ZCoT, DEF, and Contextual Augmentations

Table 8 compares the prompt methods used for multi-class classification. The ZCoT (Kojima et al., 2022) method adds the phrase "Let’s think step by step" to guide inference, applying a straightforward prompt that detects labels directly in multi-class settings. In the original DEF (Hong et al., 2024) method, when a fallacy sentence is given, the model is provided with the fallacy class to which the sentence belongs, along with the definition of that class, and is then tasked with determining whether the sentence contains a fallacy. This approach is designed for binary classification. However, for multi-class classification, providing all fallacy classes and definitions upfront can result in a form of cheating, so it cannot be directly applied. To address this, we modify the prompt for multi-class classification by providing all the logical fallacy classes in the dataset along with their definitions, instructing the model to classify the type of fallacy present in the sentence.

Additionally, Sourati et al. (2023)’s approach uses ChatGPT to generate multiple variations of each contextual augmentation (e.g., counterargument, explanation, and goal), which are manually revised, while our fully automated method employs gpt-3.5-turbo-instruct to generate a single output without manual adjustment. Their prompts solely focus on these three perspectives (counterargument, explanation, goal), whereas ours not only incorporates these perspectives but also

explicitly includes fallacy classes, leading to more effective logical fallacy detection (see Table 9).

## B.2 Example about Our Four-Step Prompts

### B.2.1 Contextual Augmentation Prompts

These prompts are utilized to generate counterarguments, explanations, and goals directly from the LLM, forming the contextual augmentation texts ( $R_i(x)$ ) as outlined in Equation 1. This foundational step, represented by  $\mathcal{R}_i(x) = \text{LLM}(x, \mathcal{I}^i)$ , allows us to delve deeper into the logical structure and validity of the argument presented in the text. Here,  $x$  represents the input text and  $\mathcal{I}^i$  signifies the instruction for generating the counterargument ( $\mathcal{I}^1$ ), explanation ( $\mathcal{I}^2$ ), or goal ( $\mathcal{I}^3$ ) of  $x$ . Fallacy classes refer to the labels assigned to each dataset that categorize the specific types of logical fallacies. Consider the [TEXT]: "Annie must like Starbucks because all girls like Starbucks."

- Counterargument

- I'll give you some texts. The texts can be question and answer pairs or sentences. The text contains one of the following logical fallacies:{fallacy classes}. Represent the counterargument to the text.
- Text: [TEXT]
- Counterargument (Example): "Not all girls like Starbucks, as personal preferences vary among individuals. Even if Annie is a girl, it does not automatically mean that she likes Starbucks. She may prefer a different type of coffee or may not like coffee at all. It is not fair to make assumptions about someone based on their gender."

- Explanation

- I'll give you some texts. The texts can be question and answer pairs or sentences. The text contains one of the following logical fallacies:{fallacy classes}. Analyze the text.
- Text: [TEXT]
- Explanation (Example): "This text suggests a generalization about girls and their preferences for Starbucks, assuming that Annie, as a girl, must also like Starbucks without evidence. This could be seen as stereotyping, making unfounded assumptions based on gender, reinforcing harmful stereotypes."

- Goal

- I'll give you some texts. The texts can

be question and answer pairs or sentences. The text contains one of the following logical fallacies:{fallacy classes}. Express the goal of the text.

- Text: [TEXT]
- Goal (Example): "The goal is to make a generalization about girls liking Starbucks based on the assumption that Annie is a girl."

### B.2.2 Reformulated Query Prompts

Following the generation of augmented texts, these prompts are employed to craft queries that delve deeper into the logical structure and validity of the arguments presented in the texts, as denoted by the equation  $\mathcal{Q}_i = \text{LLM}(x, \mathcal{R}_i)$ . In this equation,  $\mathcal{R}_i$  represents the contextual augmentation obtained from the input text  $x$ . These generated queries are intended to enhance the depth of analysis of  $x$ , improving the detection of logical fallacies. The indices  $i = 1, 2, 3$  correspond to the different aspects of queries: Counterargument ( $\mathcal{Q}_1$ ), Explanation ( $\mathcal{Q}_2$ ), and Goal ( $\mathcal{Q}_3$ ). Below are the detailed prompts for generating these queries based on the augmented texts:

- Query Generation for Counterargument Text

- I'll give you some texts and their counterarguments. The texts can be question and answer pairs or sentences. Create one query for each text to analyze the text based on its counterarguments.

- Text: [TEXT]
- Counterargument: [COUNTERARGUMENT]
- Query (Example): "How does the counterargument challenge the assumption that all girls like Starbucks?"

- Query Generation for Explanation Text

- I'll give you some texts and their explanations. The texts can be question and answer pairs or sentences. Create one query for each text to analyze the text based on its explanations.

- Text: [TEXT]
- Explanation: [EXPLANATION]
- Query (Example): "How does this text perpetuate harmful gender stereotypes and restrict individual expression?"

- Query Generation for Goal Text

- I'll give you some texts and their goals.

**The texts can be question and answer pairs or sentences. Create one query for each text to analyze the text based on its goal.**

► **Text:** [TEXT]

► **Goal:** [GOAL]

► **Query (Example):** "What does this text reveal about the speaker's attitude towards girls and their preferences?"

### B.2.3 Confidence Score Calculation for Queries

In our experiments aimed at classifying logical fallacies, we calculate the confidence scores for each query as described in Step 3. This process evaluates the probability  $p_{LLM}$  of a fallacy's class label ( $l$ ) for a given input text  $x$  based on the generated query  $Q_i$ . The confidence score for each query is computed by summing the log probabilities of the predicted tokens, which indicates how confidently the model predicts the most probable class based on the query type. These scores are then used for ranking and classification in subsequent steps.

Each **formulated prompt** involves one of three query types: Counterargument, Explanation, and Goal. Below, we provide a detailed explanation of the prompts used in this analysis:

- **Logical Fallacy Multi-Class Classification**

► Your task is to classify the type of fallacy in the Text. The label can be 'Appeal to Emotion', 'Faulty Generalization', 'Red Herring', 'Ad Hominem', and 'Irrelevant Authority'. Please classify the type of fallacy in the Text based on the Query.

► **Text:** [TEXT]

► **Formulated Prompt:** [ONE OF THREE QUERIES]

► **Label:**

### B.2.4 Prompt Ranking-Based Classification

To implement prompt ranking, we append the following instruction to the logical fallacy classification prompt: "Return only the name of the label, and nothing else. MAKE SURE your output is one of the {n\_labels}<sup>3</sup> labels stated." This instruction ensures concise outputs, preventing unnecessary information that could reduce the confidence score. We calculate the confidence scores of the results from each generated query

<sup>3</sup>{n\_labels} represents the number of labels specific to the dataset.

by using the log probabilities of the tokens outputted by LLMs (e.g., gpt-3.5-turbo, gpt-4, Llama2-7b-hf, Llama2-13b-hf, and Llama3-8B). These confidence scores are then used to rank queries in descending order, prioritizing those with higher confidence.

For example, in the Ranking Information provided below, the order of query names is based on their confidence scores: Explanation Query, Goal Query, and Counterargument Query. In this case, the Explanation Query has the highest confidence score, contributing most to the final classification. While the content of each query ( $Q_i$ ) is used in the model's input, the ranking information, which is derived by sorting the confidence scores of these queries, is incorporated into the final prompt, thereby improve the overall reliability of the fallacy classification process.

It is important to note that for LLaMA-based models, the instruction "Return only the name of the label, and nothing else. MAKE SURE your output is one of the {n\_labels} labels stated." is not always effectively applied. Therefore, for these models, we extract the log probabilities corresponding to the fallacy class in the generated output and sum these values to calculate the confidence scores.

- **Logical Fallacy Multi-Class Classification**

► Given a sentence with a logical fallacy, we aim to detect it using queries based on multiple perspectives, such as counterargument, explanation, and goal. The ranking information indicates the order of queries based on their confidence scores, which are helpful in identifying the specific type of logical fallacy present in the sentence. The label can be 'Appeal to Emotion', 'Faulty Generalization', 'Red Herring', 'Ad Hominem', or 'Irrelevant Authority'. Based on the ranking information of these queries, please reference them to detect the fallacy in the sentence.

► **Text:** [TEXT]

► **Formulated Prompt:**

- **Counterargument Query:** How does the counterargument challenge the assumption that all girls like Starbucks?

- **Explanation Query:** How does this text perpetuate harmful gender stereotypes and restrict individual expression?

- **Goal Query:** What does this text reveal

LLM	Method	PROPAGANDA		ARGOTARIO		LOGIC		COVID-19		CLIMATE	
		ACC	F1								
Llama2-7b-hf	Zero-shot	0.08	0.04	0.20	0.16	0.14	0.07	0.07	0.05	0.23	0.10
	ZCoT (Kojima et al., 2022)	0.10	0.05	0.15	0.12	0.15	0.07	0.14	0.06	0.13	0.03
	DEF (Hong et al., 2024)	0.10	0.05	0.26	0.08	0.15	0.07	0.14	0.06	0.13	0.03
	Counterargument	0.10	0.05	0.25	0.20	0.20	0.16	0.21	0.14	0.14	0.10
	Explanation	0.15	<b>0.08</b>	<b>0.36</b>	<b>0.29</b>	<b>0.37</b>	<b>0.32</b>	<b>0.29</b>	<b>0.24</b>	<b>0.41</b>	<b>0.32</b>
	Goal	0.13	0.07	0.35	<b>0.29</b>	0.31	0.26	<b>0.29</b>	0.16	0.34	0.24
	Prompt Ranking (Ours)	<b>0.21</b>	0.06	0.26	0.20	0.23	0.19	<b>0.29</b>	0.13	0.24	0.16
Llama2-13b-hf	Zero-shot	0.08	0.03	0.22	0.18	0.16	0.08	0.14	0.03	0.17	0.05
	ZCoT (Kojima et al., 2022)	0.13	0.05	0.22	0.18	0.16	0.08	0.07	0.05	0.14	0.05
	DEF (Hong et al., 2024)	<b>0.29</b>	0.05	0.27	0.17	0.18	0.05	0.07	0.02	0.24	0.08
	Counterargument	0.14	0.07	0.25	0.19	0.21	0.14	0.29	0.15	0.21	0.10
	Explanation	0.22	<u>0.10</u>	<b>0.43</b>	<b>0.34</b>	<b>0.38</b>	<b>0.32</b>	0.21	0.11	<b>0.35</b>	<b>0.26</b>
	Goal	0.20	0.08	<b>0.40</b>	<b>0.32</b>	0.37	0.27	<b>0.36</b>	<b>0.29</b>	0.28	0.20
	Prompt Ranking (Ours)	<b>0.34</b>	0.08	0.27	0.20	0.26	0.21	0.14	0.03	0.29	0.19
Llama3-8B	Zero-shot	0.13	0.05	0.21	0.13	0.25	0.13	0.07	0.02	0.14	0.08
	ZCoT (Kojima et al., 2022)	0.23	0.04	0.19	0.13	0.20	0.09	0.07	0.07	0.13	0.06
	DEF (Hong et al., 2024)	<b>0.28</b>	0.06	0.25	0.13	0.23	0.11	0.07	0.03	0.14	0.07
	Counterargument	0.13	0.06	0.34	0.26	0.29	0.20	0.21	0.21	0.15	0.11
	Explanation	0.20	<b>0.09</b>	0.51	<b>0.42</b>	<b>0.54</b>	<b>0.50</b>	<b>0.36</b>	0.24	<b>0.41</b>	0.37
	Goal	0.20	<b>0.09</b>	<u>0.53</u>	0.41	0.51	0.45	0.29	0.21	0.40	0.33
	Prompt Ranking (Ours)	0.22	<b>0.09</b>	0.30	0.29	0.38	0.36	<b>0.36</b>	<b>0.25</b>	0.33	<b>0.40</b>

Table 6: Accuracy and Macro-F1 scores of the Multi-class fallacy classification on all datasets. This table presents the experimental results for three models in the LLaMA series. **Bold**: the highest score for each model, **bold**: the highest score across all models for each dataset. **ZCoT**: Zero-shot Chain of Thought (Kojima et al., 2022), and **DEF**: the method used by Hong et al. (2024) for providing definitions of logical fallacies.

about the speaker’s attitude towards girls and their preferences?

► **Ranking Information:** Explanation Query, Goal Query, Counterargument Query  
 ► **Label:**

## C Dataset Preparation

### C.1 Fallacy Class

To ensure coherence in our analysis, we group similar fallacy classes across the datasets: **Hasty Generalization** and **Faulty Generalization** into **Faulty Generalization**; **Fallacy of Credibility**, **False Authority**, and **Appeal to Authority** into **Irrelevant Authority**; and **False Cause**, **False Causality**, **Post Hoc**, and **Causal Oversimplification** into **False Causality**.

### C.2 Data Description

For both the supervised and unsupervised settings, the datasets are split into training, validation, and test sets in proportions of 65%, 15%, and 20%, respectively, except for the LOGIC dataset, where we use the predefined test set from Jin et al. (2022). The PROPAGANDA dataset, pre-processed by Al-hindi et al. (2022), is also shuffled and divided

into training, validation, and test sets following the same 65%, 15%, and 20% split.

## D Multi-Class Classification Results on LLaMA Models

Table 6 presents the results of multi-class fallacy classification using three LLaMA models (Llama2-7b-hf, Llama2-13b-hf, and Llama3-8B). Unlike the gpt-3.5-turbo and gpt-4 results (Table 2), the Explanation (EX) query consistently shows the best performance in these models, often surpassing Prompt Ranking (PR). For example, in the Llama2-13b-hf and Llama3-8B models, Explanation scores the highest Macro-F1 and accuracy in datasets like LOGIC and ARGOTARIO. This contrasts with the larger GPT models, where Prompt Ranking leads to the most significant improvements. We hypothesize that this difference arises due to the smaller parameter sizes of the LLaMA models we used, which may limit Prompt Ranking’s effectiveness with complex query combinations. Nonetheless, PR still performs competitively, especially in the PROPAGANDA dataset, demonstrating that ranking-based methods can offer considerable benefits even in smaller models.

## E Additional Calibration Results

As shown in Section 5.2, our prompt ranking method (PR) demonstrates better calibration than the Base method, though the individual calibration performance of each query type is not fully detailed. This section provides the calibration results for each query type, offering further insight into their contributions to model reliability.

Across all datasets, PR consistently shows the greatest calibration improvements, reducing the gap between predicted confidence and actual accuracy more effectively than the Base method or individual queries. Among the queries, Explanation (EX) generally performs the best, followed by Goal (GO) and Counterargument (CG), though they do not match the overall improvement achieved by PR.

The additional results are shown in Figures 9 to 13, with an overall comparison between Base and PR across all datasets in Figure 8, highlighting the performance of Base, ZCoT (Kojima et al., 2022), DEF (Hong et al., 2024), and individual queries.

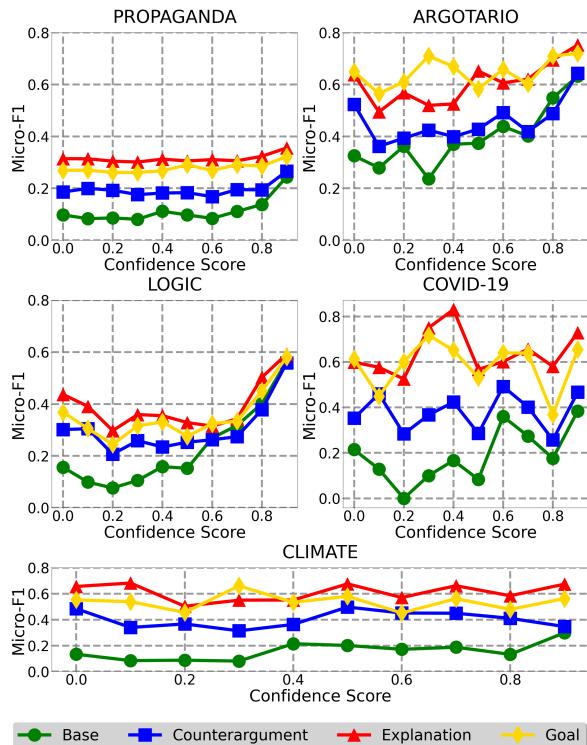


Figure 7: Relationship between confidence scores and performance with/without reformulated queries for all datasets.

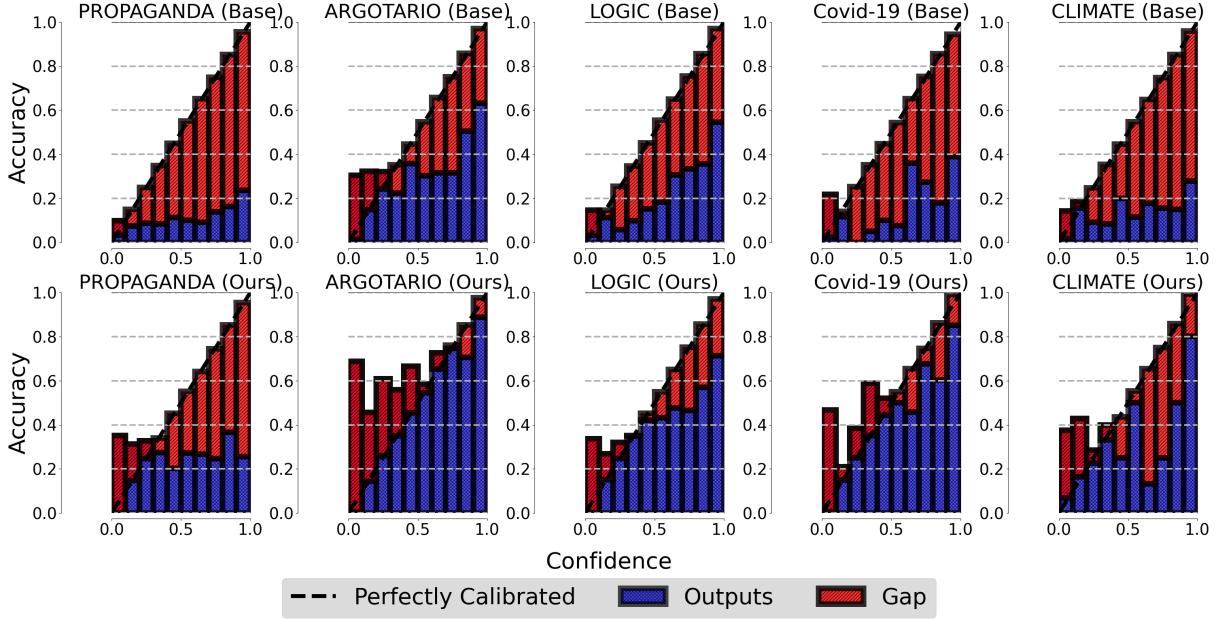


Figure 8: Reliability diagrams comparing the calibration performance of the Base method (without reformulated queries) and Ours (Prompt Ranking) across five datasets, both using the gpt-3.5-turbo model.

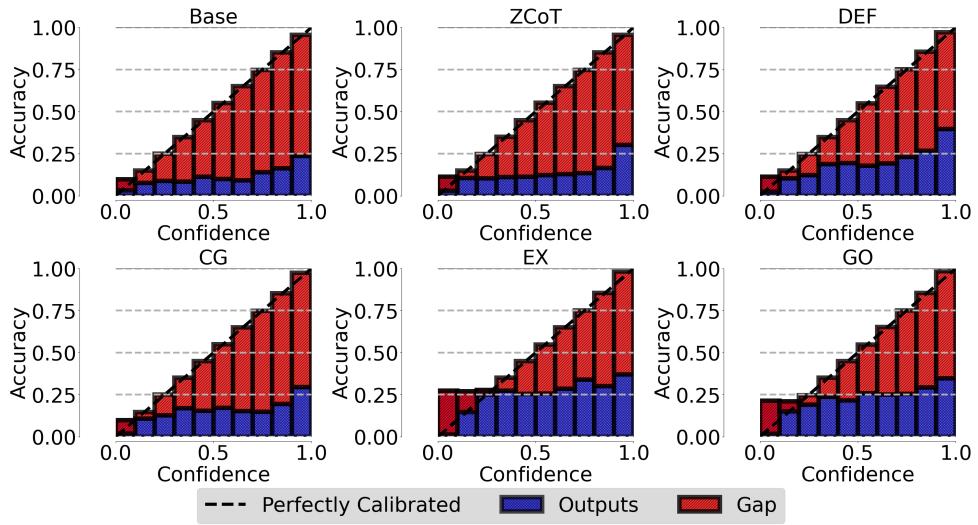


Figure 9: Reliability diagrams comparing the calibration performance of six different methods: Base, ZCoT, DEF, and three reformulated query methods—CG (Counterargument), EX (Explanation), and GO (Goal)—on the PROPAGANDA dataset.

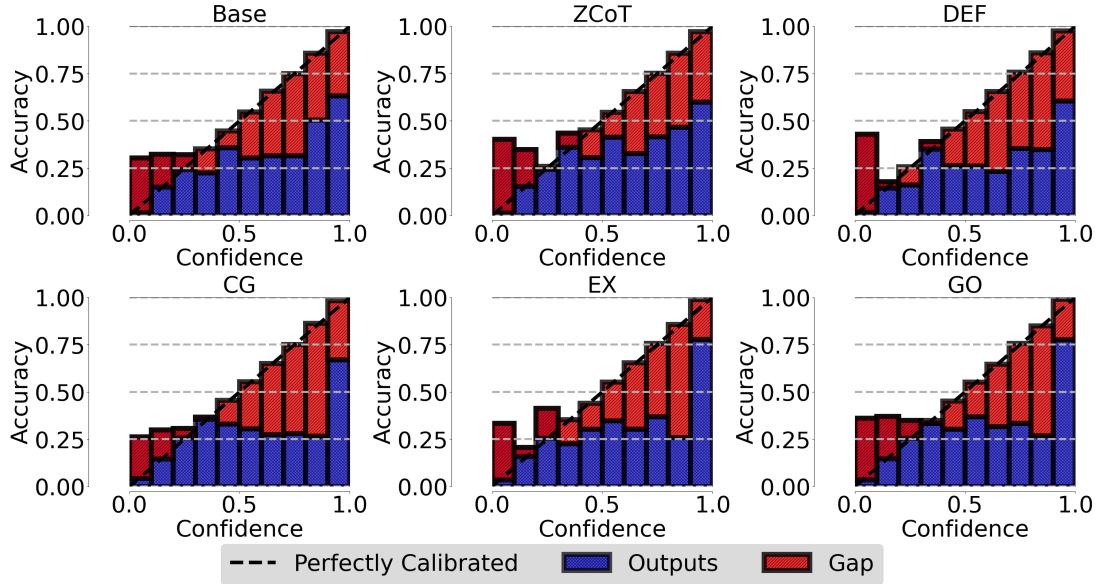


Figure 10: Reliability diagrams comparing the calibration performance of six different methods: Base, ZCoT, DEF, and three reformulated query methods—CG (Counterargument), EX (Explanation), and GO (Goal)—on the ARGOTARIO dataset.

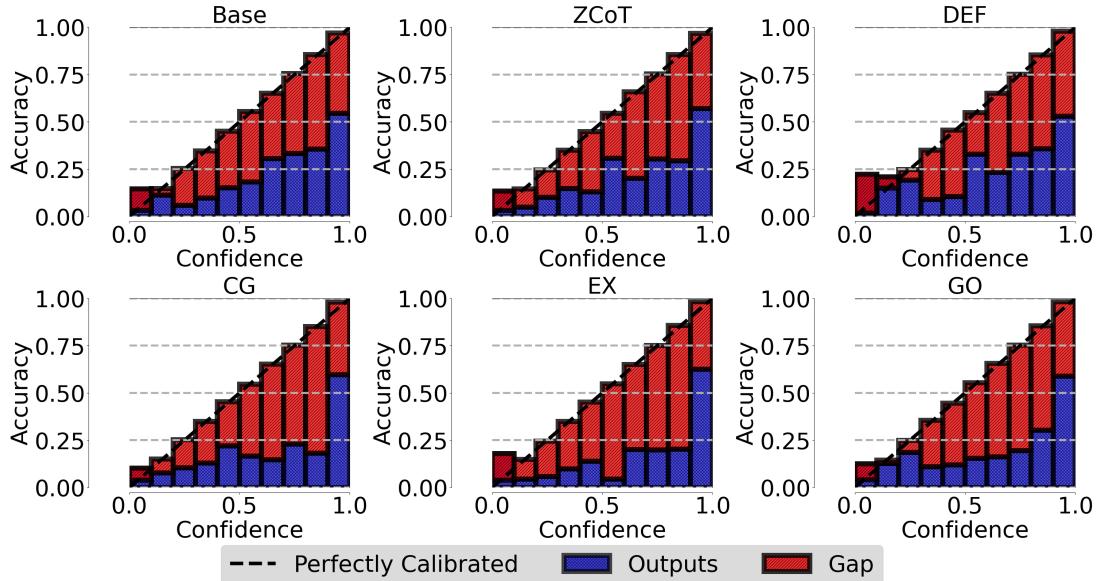


Figure 11: Reliability diagrams comparing the calibration performance of six different methods: Base, ZCoT, DEF, and three reformulated query methods—CG (Counterargument), EX (Explanation), and GO (Goal)—on the LOGIC dataset.

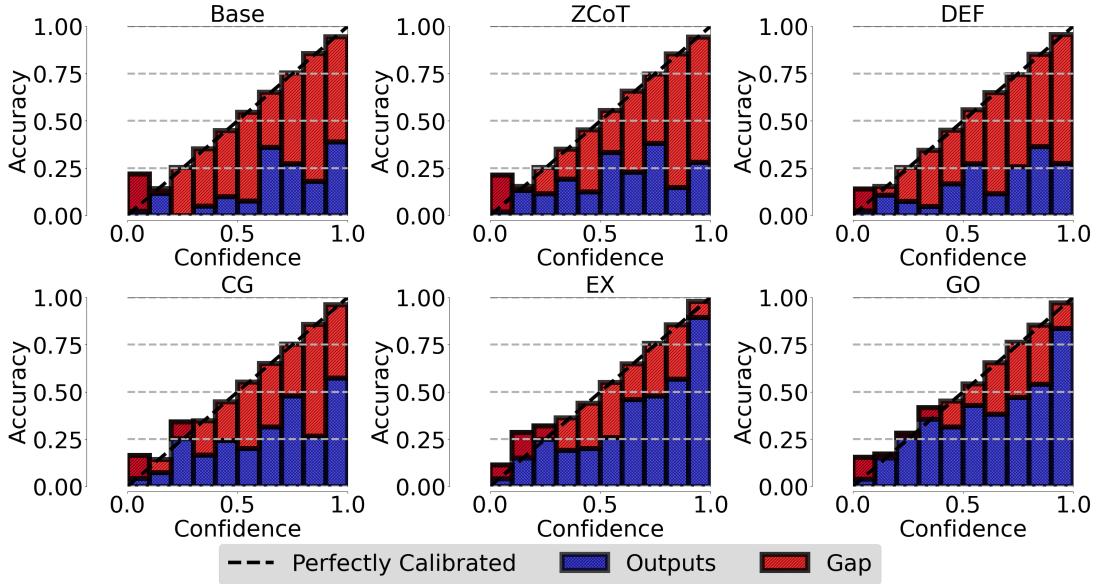


Figure 12: Reliability diagrams comparing the calibration performance of six different methods: Base, ZCoT, DEF, and three reformulated query methods—CG (Counterargument), EX (Explanation), and GO (Goal)—on the COVID-19 dataset.

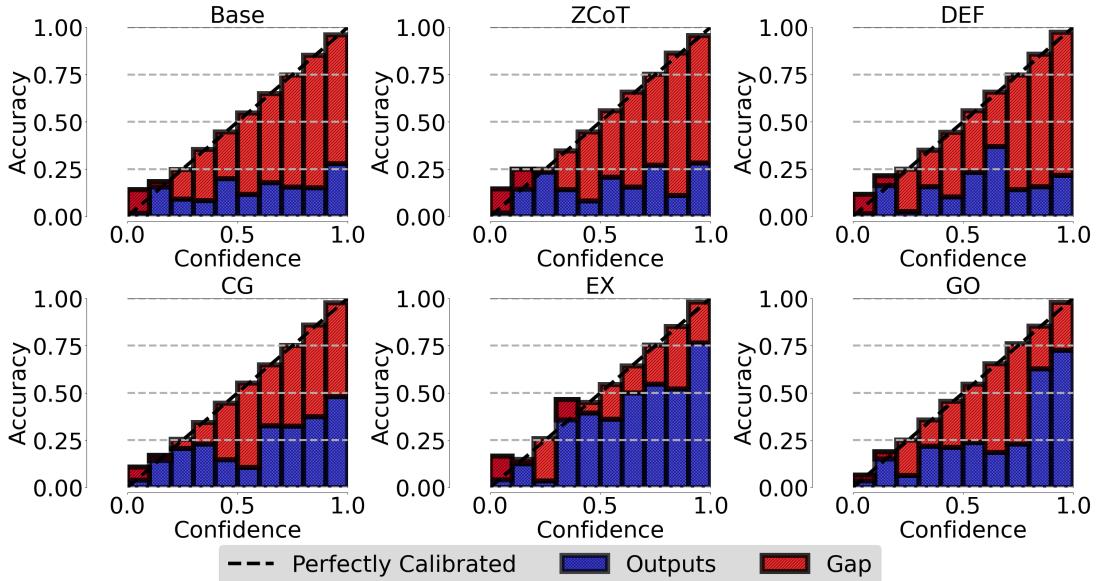


Figure 13: Reliability diagrams comparing the calibration performance of six different methods: Base, ZCoT, DEF, and three reformulated query methods—CG (Counterargument), EX (Explanation), and GO (Goal)—on the CLIMATE dataset.

Method	PROPAGANDA		ARGOTARIO		LOGIC		COVID-19		CLIMATE	
	ACC	F1								
Base	0.12	0.07	0.50	0.41	0.39	0.25	0.29	0.14	0.18	0.13
Contextual Augmentation ( <a href="#">Sourati et al., 2023</a> )										
+ Counterargument	0.13	0.08	0.48	0.40	0.35	0.20	0.21	0.13	0.12	0.07
+ Explanation	0.23	0.11	0.44	0.42	0.37	0.22	0.29	0.11	0.22	0.15
+ Goal	0.19	0.11	0.49	0.41	0.35	0.21	0.29	0.15	0.18	0.12
Reformulated Query based on <a href="#">Sourati et al. (2023)</a>										
+ Counterargument	0.12	0.07	0.45	0.37	0.35	0.21	0.29	0.15	0.20	0.13
+ Explanation	0.21	0.09	0.47	0.37	0.35	0.21	0.29	0.16	0.25	0.17
+ Goal	0.16	0.08	0.47	0.39	0.34	0.21	0.43	0.26	0.25	0.20
Contextual Augmentation (Ours)										
+ Counterargument	0.19	0.11	0.53	0.52	0.39	0.22	0.21	0.09	0.23	0.15
+ Explanation	0.20	0.10	0.51	0.48	0.41	0.24	0.29	0.22	0.25	0.12
+ Goal	0.28	0.14	0.50	0.48	0.36	0.20	0.36	0.22	0.22	0.12
Reformulated Query from (Ours)										
+ Counterargument	0.24	0.12	0.50	0.41	0.40	0.25	0.43	0.29	0.63	0.54
+ Explanation	<b>0.33</b>	0.15	0.70	0.67	<b>0.46</b>	<b>0.32</b>	0.57	0.44	<b>0.70</b>	<b>0.65</b>
+ Goal	<b>0.33</b>	<b>0.16</b>	<b>0.74</b>	<b>0.72</b>	0.44	<b>0.32</b>	<b>0.64</b>	<b>0.47</b>	0.58	0.49

Table 7: Performance comparison based on different contextual augmentation prompt methods and their resulting reformulated queries across five datasets using the gpt-3.5-turbo model. The prompt used to generate reformulated queries is the same for all cases. F1: Macro-F1 score.

Method	Prompt
ZCoT ( <a href="#">Kojima et al., 2022</a> )	Your task is to detect a fallacy in the Text. The label can be ‘Appeal to Emotion’, ‘Faulty Generalization’, ‘Red Herring’, ‘Ad Hominem’, and ‘Irrelevant Authority’. Please detect a fallacy in the Text. <b>Let’s think step by step.</b>
DEF ( <a href="#">Hong et al., 2024</a> )	Your task is to detect a fallacy in the Text. The label can be ‘Appeal to Emotion’, ‘Faulty Generalization’, ‘Red Herring’, ‘Ad Hominem’, and ‘Irrelevant Authority’. 1. Appeal to Emotion: This fallacy tries to arouse non-rational sentiments within the intended audience in order to persuade. 2. Faulty Generalization: The argument uses a sample which is too small, or follows falsely from a sub-part to a composite or the other way round. 3. Red Herring: This argument distracts attention to irrelevant issues away from the thesis which is supposed to be discussed. 4. Ad Hominem: The opponent attacks a person instead of arguing against the claims that the person has put forward. 5. Irrelevant Authority: While the use of authorities in argumentative discourse is not fallacious inherently, appealing to authority can be fallacious if the authority is irrelevant to the discussed subject. Please detect a fallacy in the Text.

Table 8: Comparison of prompt methods for fallacy classification using ZCoT ([Kojima et al., 2022](#)) and DEF ([Hong et al., 2024](#)) approaches. For ZCoT, the prompt “Let’s think step by step” is added. DEF uses definitions for each class, which are included in the prompt. The fallacy classes in the DEF prompt are from the ARGOTARIO dataset. If a query is included, the last sentence should be modified to “Please detect a fallacy in the text based on the query.”

<b>Method</b>	<b>Perspective</b>	<b>Prompt</b>
<i>CA (Sourati et al., 2023)</i>	CG	Represent the counterargument to the text.
	EX	Analyze the text.
	GO	Express the goal of the text.
<i>CA (Ours)</i>	CG	I'll give you some texts. The texts can be question and answer pairs or sentences. The text contains one of following logical fallacies: {fallacy_class} . Represent the counterargument to the text.
	EX	I'll give you some texts. The texts can be question and answer pairs or sentences. The text contains one of following logical fallacies: {fallacy_class}. Analyze the text.
	GO	I'll give you some texts. The texts can be question and answer pairs or sentences. The text contains one of following logical fallacies: {fallacy_class}. Express the goal of the text.
<i>Reformulated Query</i>	CG	I'll give you some texts and text's counterarguments. The texts can be question and answer pairs or sentences. Create one question for each text that analyzes the text based on counterarguments rather than directly asking what logical fallacy is.
	EX	I'll give you some texts and text's explanations. The texts can be question and answer pairs or sentences. Create one question for each text that analyzes the text based on explanations rather than directly asking what a logical fallacy is.
	GO	I'll give you some texts and text's goals. The texts can be question and answer pairs or sentences. Create one question for each text that analyzes the text based on goals rather than directly asking what a logical fallacy is.

Table 9: Comparison of prompt methods for contextual augmentations (CA) and reformulated queries, illustrating the approaches by [Sourati et al. \(2023\)](#) and our method. Both CA and reformulated queries are generated using the gpt-3.5-turbo-instruct model. CG: Counterargument, EX: Explanation, and GO: Goal. {fallacy\_class} refers to the classes in the dataset to which the sample (sentence) belongs.