A Logical Fallacy-Informed Framework for Argument Generation

Anonymous ACL submission

Abstract

Despite the remarkable performance of large language models (LLMs), they still struggle with generating logically sound arguments, resulting in potential risks such as spreading misinformation. An important factor contributing to LLMs' suboptimal performance in generating coherent arguments is their oversight of logical fallacies. To address this issue, we introduce fallacy-informed preference optimization (FIPO) that helps steer LLMs toward generating logically sound arguments. FIPO includes a classification loss to capture the fine-grained information on fallacy types. Our results on argument generation tasks show that FIPO reduces the fallacy errors by up to 17.5%. Furthermore, our human evaluation results reveal that the quality of the arguments generated by our method significantly outperforms the fine-tuned baselines and other preference optimization methods, such as DPO. These findings highlight the importance of ensuring models are aware of logical fallacies for effective argument generation.

1 Introduction

Argument generation is crucial in daily life and has numerous online and offline applications. For instance, legislative bodies often use persuasive arguments to secure the necessary votes for bills to pass. However, generating logically coherent arguments is a challenging task and requires an appropriate combination of *reliable evidence* and *effective logical reasoning* (Walton et al., 2008; Wachsmuth et al., 2017). Humans are prone to misconstruing logical argumentation in the real world and often unknowingly adopt flawed reasoning in discussions (Evans, 2002). Similarly, large language models (LLMs) have demonstrated limitations in their logical reasoning capabilities, suffering from logical inconsistencies (Chen et al., 2023b; Jin et al., 2022; Sourati et al., 2023), and producing logically incorrect arguments (Chen et al., 2023a).

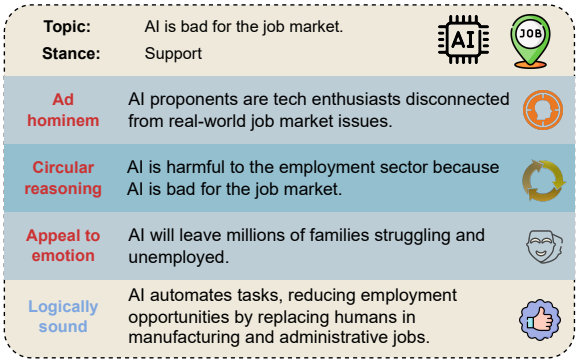


Figure 1: Examples of fallacious and logically sound arguments.

In this work, we hypothesize that LLMs generate logically incorrect arguments because they lack an understanding of logical fallacies. A logical fallacy is an error in reasoning that undermines the validity of an argument (Tindale, 2007). For example, *"I've never had the flu because I take my vitamins every day."* is an instance of a *false causality* fallacy. These fallacies arise from unsound premises. They can be identified by the absence of legitimate and relevant evidence to support their claims.

In a preliminary study, we evaluate 100 arguments on different topics with ChatGPT and find that 21% of the arguments contain fallacies. We observe that several types of logical fallacy arguments, especially *false causality and faulty generalization* are commonly generated by different LLMs such as Llama-2 (Touvron et al., 2023) or Mistral (Jiang et al., 2023). Our study explores the relationship between logical fallacy understanding and argument generation. We view models generating logically coherent arguments as a logical alignment problem, i.e., aligning the model responses (arguments) to the given topic and stance. Recent methods, such as Reinforcement Learning with Human or AI Feedback, have been shown to bridge the "alignment gap" between model responses and human preferences (Lee et al., 2023). However, LLMs

fine-tuned with RLHF can still generate arguments that are logically fallacious. Hence, to train models to prefer logically correct arguments, it is important to have reliable and diverse error scenarios as training examples. To address this issue, we define 13 categories of logical fallacy errors, drawing inspirations from the history of logic and logical fallacies (Aristotle, 2006; Russell, 2013). Figure 1 depicts some fallacy examples, and Figure 3 shows different fallacy categories.

We use ChatGPT to collect 7,872 fallacy arguments spanning different fallacy categories to train preference models. First, we perform supervised fine-tuning (SFT) to teach models to generate arguments. Next, we use preference optimization methods to instil the ability to generate logically correct arguments. Specifically, we use DPO (Rafailov et al., 2023), PPO (Schulman et al., 2017), KTO (Ethayarajh et al., 2024), and CPO (Xu et al., 2024) on our preference dataset. These fallacy-informed models demonstrate a notable improvement in argument quality, achieving a higher *win-rate* (i.e., the proportion of wins over the SFT baseline in terms of argument quality) and reducing the *fallacy-rate* (i.e., the proportion of fallacies generated) by up to 8.5%. One key challenge with the above preference learning methods is that they fail to account for the nuanced differences between logical fallacies, instead treating each fallacy similarly. To this end, we introduce **F**allacy-**I**nformed **P**reference **O**ptimization (FIPO) that combines the original preference optimization loss with a classification loss in the form of a weighted cross-entropy loss, which penalizes the model more when fallacies from the preference dataset are incorrectly classified during the training phase. We observe that FIPO outperforms the SFT baselines by reducing the *fallacy-rate* from 34.5% to 17% for Llama-2 (7B) and from 32.5% to 19.5% for Mistral (7B). FIPO also outperforms the best preference optimization method (PPO-Llama 2 and KTO-Mistral) by producing 9% and 8.25% fewer logical fallacy errors respectively.

Contributions: (i) To the best of our knowledge, we are the first to study and show how understanding logical fallacies can improve argument generation quality. (ii) We introduce FIPO, which integrates a classification loss during the preference optimization phase, which helps further reduce the *fallacy-rate*. (iii) Human evaluation results show that FIPO generates higher-quality arguments than other preference learning methods and performs well on out-of-domain scenarios, indicating its ability to generalize to unseen topics. These findings underscore the importance of making models aware of logical fallacies for argument generation. (iv) Our code is available here (anonymized), and we hope our study inspires further research on logical fallacies in argumentation.

2 Related Work

Logical Fallacies. Logical fallacies are errors in reasoning that can undermine the validity of an argument (Tindale, 2007). In argumentative discourse, identifying fallacies is crucial for measuring the quality of argumentation (Wachsmuth et al., 2017; Nakpih and Santini, 2020). Prior works have shown that LLMs struggle to classify logical fallacies, with F1 scores reaching 66% (Jin et al., 2022; Sourati et al., 2023). More recently, Li et al. (2024) demonstrated GPT-4's ability to identify and classify fallacies, achieving over 86% accuracy in both tasks. Nevertheless, previous works have not explored how a nuanced understanding of logical fallacies might influence argument generation.

Argument Generation. Argument generation is an important task in natural language processing that involves generating coherent and persuasive arguments for a given topic. Existing argument generation frameworks have made significant strides: Hua and Wang (2018) introduced a generator that creates arguments from key phrases, followed by a separate decoder to produce the final argument text. Hua et al. (2019) developed Candela, a style-controlling counter-argument generation framework. Schiller et al. (2021) presented Arg-CTRL, a model that uses control codes for topic, stance, and aspect in sentence-level argument generation. More recently, Saha and Srihari (2023) introduced an argument generator for factual arguments across a limited set of topics. Despite these advances, no study has yet addressed generating arguments from the lens of logical fallacies.

Data Generation and Automatic Evaluation with LLMs. Using LLMs like ChatGPT in data generation is supported by their proven effectiveness in a broad spectrum of text generation tasks, including the creation of instructional and relational datasets (Peng et al., 2023; Sun et al., 2023; Wang et al., 2023; Shao et al., 2023; Chia et al., 2022). Notably, Schick and Schütze (2021) demonstrated the utility of LLMs in producing datasets that sig-

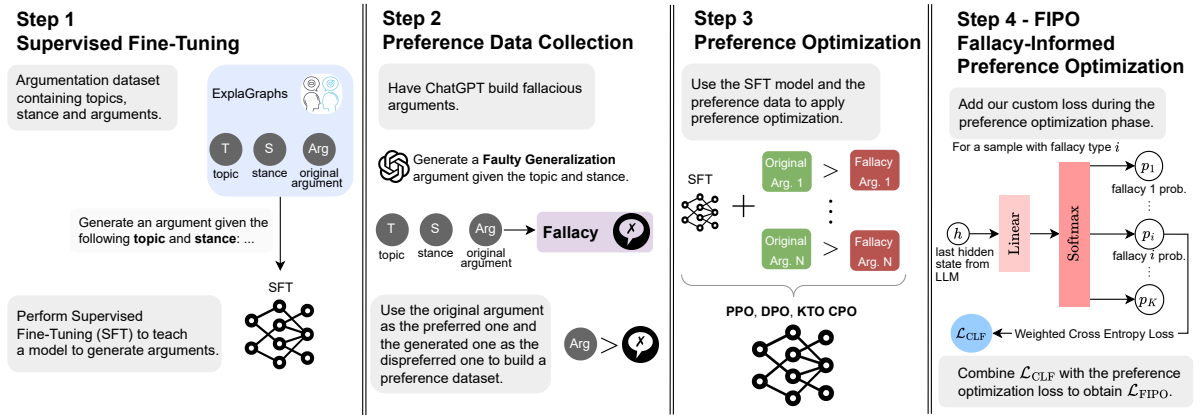


Figure 2: Overview of our framework. The first step is supervised fine-tuning using argumentation data. Next, we collect preference data by generating fallacious arguments using ChatGPT. We then perform preference optimization using methods like DPO, PPO, CPO, and KTO. Finally, we introduce FIPO, which integrates a classification loss during the preference optimization phase.

nificantly enhance the training of smaller models. Regarding the evaluation of automatically generated text, Liu et al. (2023) highlight that traditional reference-based metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) are inadequate for tasks requiring high creativity and diversity. Motivated by this, we prioritize GPT-4's automated evaluation capabilities to directly assess the effectiveness of our alignment methods within the preference dataset. Given Li et al. (2024)'s demonstration of GPT-4's ability to identify and classify logical fallacies, we use it as a judge to identify fallacies in arguments. We validate GPT-4's reliability in classifying fallacies by performing a human annotation task.

3 Task Formulation

In this work, we address the argument generation task, where a model is required to generate a logically correct argument given a topic and a stance. We denote the dataset as $\mathcal{D} = \{t^{(i)}, s^{(i)}, y_w^{(i)}\}_{i=1}^{N}$, where t represents the input topic, s the stance (either supporting or counter), and y_w denotes the argument. N is the number of instances. One naive approach to address the problem of logical argument generation is prompting LLMs. To assess this approach, we evaluate GPT-3.5, Llama-2 (7B), and Mistral (7B) in the zero-shot setting on a set of 100 topics. Additionally, we implement a Retrieval Augmented Generation (RAG) model on top of Llama-2 using the `wiki-dpr` database (Karpukhin et al., 2020). A comprehensive description and example are provided in Appendix C.2. **We observe models struggle to generate logically sound arguments.** In Tab. 1, we report the performance of all the models in argument generation. We use GPT-4 to assess the *fallacy-rate* of the generated arguments. Among the models evaluated, ChatGPT demonstrated the best performance; however, it still generates fallacious arguments in 21% of the cases. Furthermore, open-source models were observed to produce higher rates of fallacious arguments. The detailed distributions of fallacy types across different approaches in the zero-shot setting are presented in Tab. 6.

Model	ChatGPT	Llama-2	Mistral	Llama-2-RAG
fallacy-rate	21	55	38	37

Table 1: *fallacy-rate* (% of fallacies) for arguments generated by different models.

While models like ChatGPT offer the potential to assist in generating arguments across a wide variety of topics, the presence of fallacies within these arguments persists.

4 Methodology

To address the challenge of generating fallacy-free and logically sound arguments, we propose employing preference learning methods to generate arguments logically aligned with the given topic and stance. This approach involves making models aware of logical fallacies and training them to generate logically correct arguments by rewarding valid arguments and penalizing dispreferred samples. The process of preference learning typically involves three main steps: (i) a supervised fine-tuning (SFT) step (Section 4.1), (ii) a preference data collection step (Section 4.2) and (iii) a reinforcement learning step (Section 4.3). In Sec-

tion 4.4, we introduce our method (FIPO), which introduces fine-grained information about fallacies in the alignment process. A comprehensive overview of the methodology is presented in Fig. 2. To justify our design and methodology for FIPO and the preference data collection, we perform two ablation studies with different training approaches, described in Appendix F. The results demonstrate that our design achieves the best performance.

4.1 Supervised Fine-Tuning

We fine-tune a pre-trained language model (π_β) on the argumentation dataset $\mathcal{D}$ and use maximum likelihood estimation to obtain a model π_{SFT}.

$$\mathcal{L}_{\text{SFT}}(\pi_\beta) = -\mathbb{E}_{(t,s,y_w)\sim\mathcal{D}}\left[\log\left(\pi_\beta(y_w|t,s)\right)\right] \quad (1)$$

4.2 Preference Data Collection

Conventionally, after the SFT phase, π_{SFT} is prompted with input x to produce pairs of outputs $(y_1, y_2) \sim \pi_{\text{SFT}}(y|x)$, which are then presented to human annotators to rank as preferred and dispreferred responses. However, our objective is to reduce logical fallacy errors in the model's outputs. Therefore, including a diverse range of fallacy types in the preference data is essential, as these may not be sufficiently represented in the model's outputs. Drawing inspiration from the history of logic and logical fallacies (Aristotle, 2006; Russell, 2013), we define 13 categories of logical fallacy errors (see Fig. 3). However, there are two key challenges: **(i)** determining the appropriate distribution of logical fallacy errors in the preference data, and **(ii)** automatically collecting such fallacy arguments.

To address the first concern, we leverage the LOGIC dataset (Jin et al., 2022), which was carefully curated through extensive web crawling and data collection from diverse online sources. This dataset reflects the distribution of fallacies in real-world scenarios, providing a realistic foundation for mitigating fallacies in everyday argumentative discourse. For the second concern, we collect fallacy arguments using ChatGPT, following the distribution of fallacies in Fig. 3. To ensure arguments generated by ChatGPT are indeed fallacies, we provide a definition of the specific fallacy being generated and examples of that fallacy type. To populate our preference dataset and ensure it spans across the most types of fallacies, we generate four fallacious arguments with different fallacy types for each sample. To emphasize certain fallacy types more than others, we sample the fallacy

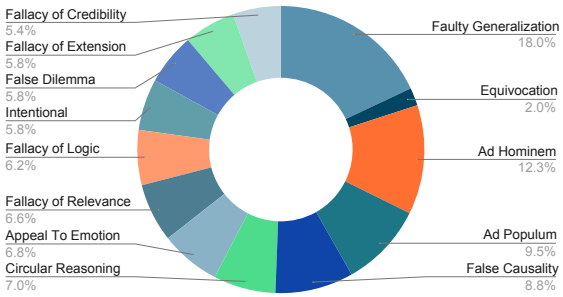


Figure 3: Distribution of different fallacy types according to the LOGIC dataset (Jin et al., 2022), based on which we generate the negative preference data.

types to generate according to the LOGIC dataset (Jin et al. (2022), Fig. 3). The original dataset $\mathcal{D}=\{t^{(i)}, s^{(i)}, y_w^{(i)}\}_{i=1}^{N}$ containing the topic stance and argument is now augmented with fallacies, denoted as y_l. We define the preference dataset as $\mathcal{D}' = \{t^{(i)}, s^{(i)}, y_w^{(i)}, y_l^{(i)}, k^{(i)}\}_{i=1}^{M}$, where we have M pairs of preferred (y_w) and dispreferred (y_l) samples, with $k^{(i)}$ being the fallacy type of the dispreferred argument $y_l^{(i)}$. Since we generate four fallacies for each argument, it ensues that $M = 4N$. The test set is not augmented with fallacies, as we use only the topics and stances at inference time to evaluate the quality and logical soundness of the arguments generated. More details on the generations and our prompt designs for ChatGPT are presented in Appendix A.1. Finally, our augmented fallacy argument dataset consists of the train-test split shown in Tab. 2.

	# Train	# Test
EXPLAGRAPHS data	1,968	400
Generated Fallacies	7,872	-
Total	7,872	400

Table 2: Train-Test split of our augmented preference dataset.

4.3 Preference Learning Phase

In this work, we use four preference learning algorithms: PPO, DPO, KTO, and CPO. Among these, only PPO requires explicit feedback from a reward model. For the other methods, we apply the preference optimization using π_{SFT} as a reference model and the preference data $\mathcal{D}'$.

Explicit Reward Modelling. We use the dataset $\mathcal{D}'$ to train the Electra model (Clark et al., 2020) to learn to predict reward values.

Implicit Reward Modeling. Methods like DPO, KTO, and CPO employ contrastive loss to derive implicit rewards from preference datasets. Note that CPO (Xu et al., 2024) is a reference-free method that does not require a reference policy. The different methods are detailed in Appendix B.

4.4 Fallacy-Informed Preference Optimization (FIPO)

Despite the preference optimization, models persistently generate specific types of logical fallacies, particularly *faulty generalization* and *false causality* arguments (Tab. 4). This can be attributed to the fact that the models do not explicitly learn about the fallacy types. Hence, we propose FIPO, which uses a classification head attached to the generative model to calculate a weighted cross-entropy loss for the preferred and dispreferred samples. Recall the preference dataset $\mathcal{D}' = \{t^{(i)}, s^{(i)}, y_w^{(i)}, y_l^{(i)}, k^{(i)}\}$ where $k \in [1, 13]$ is the fallacy type of y_l. We also label the preferred samples y_w as '*Not a Fallacy*'. Secondly, after a forward pass through the language model π_θ, we extract the hidden state from the last hidden layer, denoted as $h_\theta^L(y|t,s)$, where L represents the number of layers in the base models. This hidden state is fed into the classification head, and the resulting output defines the probability for fallacy type k:

$$p_k\big(h_\theta^L(y|t,s)\big) = \text{Softmax}\big(W h_\theta^L(y|t,s) + b\big)_k \tag{2}$$

where W is the linear layer's weight matrix, and b is the corresponding bias term. To avoid penalizing the model equally for misclassifying different types of fallacies, we propose to guide the model to prioritize the most frequent fallacy types. This approach ensures that the model accurately identifies the most occurring fallacies. We define weights w_k for each fallacy type k as its frequency in $\mathcal{D}'$: $w_k = \frac{1}{|\mathcal{D}'|}\sum_{i=1}^{|\mathcal{D}'|} \mathbb{1}\{k^{(i)} = k\}$ where $\mathbb{1}$ is the indicator function. The fallacy-informed classification loss is defined as a weighted cross-entropy loss:

$$\mathcal{L}_{\text{CLF}}(\pi_\theta) = -\mathbb{E}_{(t,s,y_w,y_l,k)\sim\mathcal{D}'} \Big[w_0 \log p_0(h_\theta^L(y_w|t,s)) + w_k \log p_k(h_\theta^L(y_l|t,s))\Big] \tag{3}$$

where $w_0 = \min_k w_k$, which is designed to let the model focus more on the fallacies in the less preferred samples rather than the non-fallacy samples during the preference optimization process. The resulting loss function, termed **F**allacy **I**nformed **P**reference **O**ptimization loss, combines the loss from the preference optimization with our classification loss ($\mathcal{L}_{\text{CLF}}$). In our work, CPO (Xu et al., 2024) is the method with which we combine our loss since it has the best trade-off between *win-rate* and *fallacy-rate* (see Fig. 4 & 5, Tab. 4). The resulting loss is:

$$\mathcal{L}_{\text{FIPO}}(\pi_\theta) = \mathcal{L}_{\text{CPO}}(\pi_\theta) + \lambda \mathcal{L}_{\text{CLF}}(\pi_\theta) \tag{4}$$

where λ is a weighting parameter to adjust the fallacy-informed loss with respect to the preference optimization loss. A more detailed description of $\mathcal{L}_{\text{FIPO}}$ is described in Appendix G and Appendix C.1 contains more information on the λ parameter tuning.

5 Experimental Setup

We denote the policies obtained after the SFT phase and the alignment phase as π_{SFT} and π_θ, respectively. The policy π_θ is aligned using one of the following methods: PPO (Schulman et al., 2017), DPO (Rafailov et al., 2023), CPO (Xu et al., 2024), KTO (Ethayarajh et al., 2024), and FIPO.

Datasets. We evaluate argument generation based on the ExplaGraphs dataset (Saha et al., 2021) where samples contain a *Topic*, a *Stance*, and short *Arguments* (5-20 words), spanning a wide range of topics. We augment this dataset by generating equivalent short arguments in the form of fallacies using ChatGPT, based on the distribution of the LOGIC dataset (Jin et al., 2022), as described in Section 4.2 and illustrated in Fig. 2. The LOGIC dataset also contains short-length fallacies. Based on the length of arguments and fallacies, our study focuses only on short argumentative texts. The final dataset size is provided in Tab. 2. We also perform out-of-domain analysis on a subset of samples from the Debatepedia dataset (Cabrio and Villata, 2012).

Settings. We use Llama-2 (7B) (Touvron et al., 2023) and Mistral (7B) (Jiang et al., 2023) as our base models. For each alignment method, we leverage Low-Rank Adaptation (LoRA) (Hu et al., 2021). This drastically reduces the number of parameters that need to be fine-tuned, from 7B to 8.3M ($\approx$0.12%). For both base models, we obtain a reference policy π_{SFT} and an aligned policy π_θ for every alignment method. More details about hyperparameters, including training and decoding parameters, are described in Appendix C.1.

Metrics. We use two metrics to evaluate the arguments generated by the aligned models compared to the baseline SFT model: the *win-rate* and the *fallacy-rate*. The *win-rate* measures the proportion of instances where one argument is judged to be of higher quality than the other, while the *fallacy-rate* represents the proportion of logical fallacies detected in the generated arguments.

Human Evaluation. We conduct a human evaluation to compute the *win-rate*, to validate the relevance and quality of the generated arguments. We select 200 samples from the set of generated arguments by the aligned models, along with the corresponding topics, stances (either supporting or counter) and the equivalent arguments generated by the SFT model. Annotators are requested to perform a comparative evaluation between the SFT and aligned models by determining which argument is superior or whether both arguments are equally good or equally bad. Despite the subjective nature of this task, as certain arguments may appeal differently to different individuals, we provide instructions to annotators including selecting the argument that most clearly addresses the topic and stance. Refer to Fig. 6 for more details. We also perform an annotation task where workers agree or disagree with GPT-4's fallacy predictions. This allows us to validate GPT-4 as a judge for computing the *fallacy-rate*. Annotators are recruited from Amazon Mechanical Turk[1] for this task. We limit our selection to native English speakers residing in the United States. The eligibility criteria for annotators include a HIT approval rate of at least 97% and a minimum of 10,000 approved HITs. We present more details, including the annotating instructions in Appendix C.4.

Automatic Evaluation with GPT-4. We perform a pairwise comparison between the samples generated by π_{SFT} and its counterparts generated by π_θ using the *win-rate* and *fallacy-rate*. For the *win-rate*, we prompt GPT-4 to decide which argument is superior, as described in Appendix A.2, or if both are equally good. For the *fallacy-rate* , we prompt GPT-4 to evaluate the argument and detect whether the argument is any of the fallacy types out of all the ones listed in Tab. 9, if one is present. A description of the GPT-4 evaluation and prompts is detailed in Appendix A.2.

[1] mturk.com

Agreement Metric	Value
Randolph's-κ	0.640
Majority agreement ratio	0.955

Table 3: Agreement scores. Randolph's κ reflects agreements between annotators. 0.64 indicates substantial agreement. The majority agreement ratio is used to compute the agreement rate between annotators and GPT-4. A majority agreement of 95.5% indicates the majority agrees with GPT-4's prediction in most cases.

GPT-4's Reliability in Detecting Fallacies. We rely on GPT-4 to compute the *fallacy-rate* Li et al. (2024) recently showed that GPT-4 performs well on logical fallacy detection, achieving 86% accuracy. To validate GPT-4's fallacy predictions, we use human annotators and ask them to *agree* or *disagree* with the predictions. This binary task is simpler and more cost-effective than asking humans to classify arguments into fallacy types. Additionally, this provides a clearer assessment of GPT-4's reliability. More details are provided in Appendix E.1.

We collect three annotations for each sample, consisting of "Agree" or "Disagree" responses. We compute the percentage agreement rate to assess the agreements between annotators and GPT-4, reflecting how often the majority vote matches GPT-4's prediction. This directly assesses GPT-4's competence in classifying fallacies, making it reliable for computing the *fallacy-rate*. We also compute the agreement among annotators using Randolph's κ (Randolph, 2005), which is well-suited for scenarios with rating imbalances (e.g., frequent "Agree" responses). Further details on the computation and interpretation of Randolph's κ are provided in Appendix E.2. The agreement rates are displayed in Tab. 3, and the results demonstrate the effectiveness and reliability of using GPT-4 for classifying fallacies.

6 Experimental Results

As outlined in Section 5, our evaluation of the generated arguments focuses on two primary aspects: (i) pairwise comparison of argument quality between the reference policy π_{SFT} and the aligned policies π_θ, which is detailed in Section 6.1; and (ii) the analysis of *fallacy-rate* across different preference optimization methods in Section 6.2.

6.1 Pairwise Comparison of Different Preference Optimization Methods

We perform a pairwise comparison to compute the *win-rate* between arguments generated by the SFT

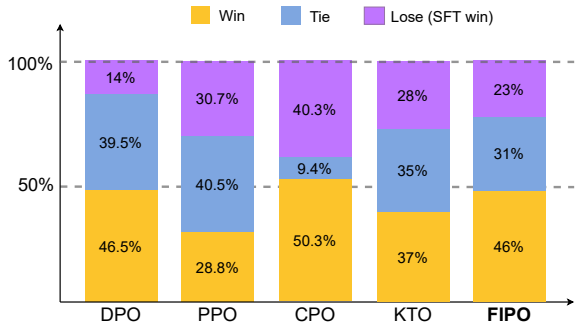


Figure 4: Human evaluation results comparing arguments generated by different preference optimization strategies using Llama-2 as the base model. The *win-rate* indicates how often each policy outperforms the SFT baseline regarding argument quality regarding the response to a given topic and stance.

baselines and all the aligned models. Each argument undergoes a manual and automatic (GPT-4-based) comparative evaluation, whose results are shown in Fig. 4 and Tab. 7. For the human evaluation, each sample receives three assessments. Samples lacking majority consensus among annotators are excluded from further analysis. From the human annotated *win-rate* depicted in Fig. 4, we address the following research questions:

RQ$_1$: Are preference optimization methods better than SFT? The aligned policies outperform π_{SFT} in terms of *win-rate*, indicating an improvement in overall argument quality. DPO, CPO and FIPO are the only methods achieving over 40% *win-rate*, demonstrating a better ability to generate qualitative arguments. π_{CPO} stands out as it has the highest percentage of wins (50.3%), followed closely by π_{DPO} (46.5%). We observe similar results in the automatic *win-rate* evaluation with GPT-4 (Fig. 5), where DPO, CPO and FIPO have the highest *win-rates*.

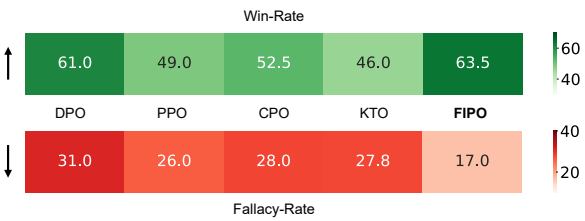


Figure 5: GPT-4 evaluation of *win-rate* and *fallacy-rate*, for arguments generated by Llama-2. The *win-rate* represents the frequency with which the aligned policy outperforms SFT (for example, DPO won 61% of the time). The *fallacy-rate* measures the proportion of the detected fallacies. The best-performing policy is our proposed FIPO method that achieves the lowest *fallacy-rate* (17%) and the highest *win-rate* (63.5%).

RQ$_2$: Does FIPO improve from existing preference methods? FIPO uses the classification loss defined in Section 4.4 on top of CPO's loss. This is because CPO achieves the best trade-off between *win-rate* and *fallacy-rate* despite having a higher loss-rate (Fig. 4). We denote this policy as π_{FIPO}. Although the *win-rate* of π_{FIPO} is slightly lower at 46% compared to CPO's 50.3%, it is essential to note the significant decrease in loss-rate—from 40.3% to 23%. This reduction indicates that π_{SFT} wins against π_{FIPO} less frequently, suggesting that π_{FIPO} produces arguments that are more qualitative compared to the baseline and to π_{CPO}. This improvement highlights the benefits of including more fine-grained details in FIPO's loss, making models more aware of logical fallacies, as FIPO now yields outcomes that are not only equivalent but often superior to those generated by the π_{SFT}. Similar observations are drawn from the automatic *win-rate* evaluation in Appendix D.2, where GPT-4 served as the judge to evaluate the arguments and only CPO achieved over 50% for both Llama-2 and Mistral, as shown in Tab. 7.

6.2 Results for *Fallacy-Rate* and Types

Evaluating text segments to identify logical fallacies poses inherent challenges for humans. Detecting such fallacies demands an extensive understanding of logical principles and argumentative structures. Without a robust grasp of logical fallacies, differentiating between valid and flawed reasoning becomes difficult. Additionally, personal biases and preconceptions can also cloud judgment, leading to overlooked fallacies or biased interpretations of arguments. We report GPT-4's evaluations in Tab. 4. Based on the results, we address the following research questions:

RQ$_3$: Do preference optimization methods mitigate logical fallacy errors? The aligned policies produce fewer fallacies compared to the SFT baselines. Specifically, every alignment method outperforms π_{SFT} for Llama-2. For Llama-2 and Mistral, DPO is the method that improves the least, and even produces more fallacies than π_{SFT} with Mistral, having a higher *fallacy-rate*. The other methods (PPO, CPO and KTO) consistently outperform SFT and produce fewer fallacies.

RQ$_4$: Does FIPO further reduce logical fallacy errors? The least fallacy producing pol-

	Llama-2 (7B)						Mistral (7B)					
Fallacy Types	**SFT**	**DPO**	**PPO**	**CPO**	**KTO**	**FIPO**	**SFT**	**DPO**	**PPO**	**CPO**	**KTO**	**FIPO**
Faulty Generalization	27.5	21	17.5	19.25	21	7	23	24	22.25	22.25	21	9.5
False Causality	2.5	5	4.25	4.75	4.5	3.5	5.25	5.75	5	4	3.5	4
Appeal To Emotion	1	1.25	0.75	1.75	-	2.5	1.25	1.75	0.25	1.5	1.75	3
Equivocation	1	1	1.25	0.25	0.75	-	0.75	-	0.5	0.25	0.25	-
Fallacy of Relevance	0.5	0.25	0.75	0.25	0.25	-	-	0.5	-	0.75	0.25	0.5
Circular Reasoning	1	-	1.25	-	0.75	1.5	0.75	0.25	-	-	-	0.5
Ad Populum	-	1.25	-	0.5	-	-	0.25	0.25	0.25	1	0.25	1
False Dilemma	1	1.25	-	1	0.25	2.5	1	1	0.5	0.25	0.75	1
Ad Hominem	-	-	0.25	0.25	0.25	-	0.25	0.25	0.25	-	-	0.5
Not A Fallacy	65.5	69	74	72	72.25	83	67.5	66.25	71	70	72.25	80.5
Fallacy-Rate ↓	34.5	31	26	28	27.75	**17**	32.5	33.75	29	30	27.75	**19.5**

Table 4: *Fallacy-rate* (in percentages) of each policy, as detected by GPT-4. We omit other fallacy types as none of them were reported by GPT-4. FIPO is the top-performing method, producing the least amount of fallacies.

icy is π_{FIPO}, achieving a *fallacy-rate* of 17% for Llama-2, outperforming the previous best of 26% (PPO). For Mistral, FIPO has a *fallacy-rate* of 19.5%, outperforming the previous best of 27.75% (KTO) (Tab. 4). More specifically, FIPO, based on CPO, beats CPO by 11% and 10.5% for Llama-2 and Mistral, respectively. This highlights the utility of the classification loss, indicating that the policies have a better understanding of logical fallacies than regular preference optimization.

RQ$_5$: What is the most observed fallacy type? The most frequently observed fallacy produced across all policies is *Faulty Generalization*. For π_{FIPO}, the occurrence for this type is only 7%, effectively integrating the concept of generalization. An example of this occurrence is illustrated in Tab. 11. As *Faulty Generalization* is the most frequent fallacy type in our preference dataset (18%, Fig. 3), the weight assigned to this fallacy type in Equation 3 is the largest. Consequently, a higher classification loss is incurred if these fallacies are misclassified, enhancing the language model's ability to accurately identify and reduce occurrences of *Faulty Generalization*. We also observe that of the 13 fallacy types, GPT-4 never classifies arguments in the following classes: *Fallacy of logic, credibility, extension* and *intentional*.

6.3 Out-of-Domain Analysis

To showcase the effectiveness of alignment methods in argument generation, we evaluate the *fallacy-rate* and *win-rate* of 100 arguments generated on a subset of Debatepedia (Cabrio and Villata, 2012). We used Llama-2 as the base model for these alignment methods. The results, presented in Tab. 5, show that π_{FIPO} is the second-best policy in terms of minimizing fallacies (55%), slightly behind π_{KTO} (56%). We also find that FIPO achieves the highest *win-rate*, winning 62% of the times against π_{SFT}. Interestingly, our results reveal that FIPO helps to reduce *False Causality* and *Fallacy of Relevance* fallacies.

Fallacy Types	**SFT**	**DPO**	**PPO**	**CPO**	**KTO**	**FIPO**
Faulty Generalization	17	20	17	24	17	18
False Causality	9	7	6	8	8	5
Appeal To Emotion	7	16	13	13	7	12
Fallacy of Relevance	12	6	6	10	3	-
Ad Populum	3	4	2	1	1	-
False Dilemma	6	6	6	4	6	7
Equivocation	-	-	2	1	1	2
Circular Reasoning	4	2	-	2	1	1
Fallacy-Rate ↓	58	61	52	63	**44**	45
Win-Rate vs. SFT ↑	-	59	54	43	55	**62**

Table 5: Fallacies generated by different alignment methods in the out-of-domain setting, detected by GPT-4. We omit the other fallacy types, as none of them were reported as such by GPT-4. We also evaluate the *win-rate* and observe that FIPO achieves the highest one and is the second-best policy at not generating fallacies.

7 Conclusion

In this work, we investigate the impact of logical fallacies on argument generation and introduce FIPO, a novel framework designed to improve the logical soundness of arguments by including a classification loss during the preference optimization phase. Both human and automatic evaluations show that our method produces higher-quality arguments and achieves lower *fallacy-rates*. These findings underscore the importance of addressing logical fallacies in improving argument generation.

Limitations

Although various preference optimization strategies have shown improvement over the SFT baseline in reducing fallacious arguments, the margin remains modest. This may be attributed to several factors: our assumption that the original dataset (Saha et al., 2021) was free of fallacies, the inherent complexity and diversity of fallacies which complicates effective detection, and the variability in model performance, particularly the weaker results from the Mistral model compared to Llama-2. Additionally, the limited size of our dataset and the brevity of arguments present further challenges, as the lack of contextual cues can hinder the models' ability to identify and avoid fallacies consistently.

Ethics Statement

In this paper, we experiment with existing datasets that are well-acknowledged. Our framework is designed to improve argument generation in LLMs, which have been shown to potentially encode biases about race, gender, and other demographic attributes (Weidinger et al., 2021; Sheng et al., 2020). Since our work does not offer a way to mitigate these biases, models could still possibly reflect the same harmful behaviors. We recommend anyone deploying our model *off-the-shelf* should first check whether the model is harmful towards any protected group, and appropriate mitigation should be taken. Moreover, our annotation task is based on arguments generated by these models, which are complex to evaluate in terms of identifying logical fallacies because of the complex nature of fallacious reasoning. Hence, our task was done with English-speaking AMT workers who were paid adequately for the time it took to solve the tasks.

References

Aristotle. 2006. *On Sophistical Refutations*. The Internet Classics Archive.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Elena Cabrio and Serena Villata. 2012. Natural language arguments: A combined approach. *Frontiers in Artificial Intelligence and Applications*, 242.

Guizhen Chen, Liying Cheng, Anh Tuan Luu, and Lidong Bing. 2023a. Exploring the potential of large language models in computational argumentation. *ArXiv*, abs/2311.09022.

Meiqi Chen, Yubo Ma, Kaitao Song, Yixin Cao, Yan Zhang, and Dongsheng Li. 2023b. Learning to teach large language models logical reasoning. *arXiv preprint arXiv:2310.09158*.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization.

Jonathan St BT Evans. 2002. Logic and human reasoning: an assessment of the deduction paradigm. *Psychological bulletin*, 128(6):978.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Xinyu Hua, Zhe Hu, and Lu Wang. 2019. Argument generation with retrieval, planning, and realization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2661–2672, Florence, Italy. Association for Computational Linguistics.

Xinyu Hua and Lu Wang. 2018. Neural argument generation augmented with externally retrieved evidence. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–230, Melbourne, Australia. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schoelkopf. 2022. Logical fallacy detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7180–7198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yanda Li, Dixuan Wang, Jiaqing Liang, Guochao Jiang, Qianyu He, Yanghua Xiao, and Deqing Yang. 2024. Reason from fallacy: Enhancing large language models' logical reasoning through logical fallacy understanding. *arXiv preprint arXiv:2404.04293*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Callistus Ireneous Nakpih and Simone Santini. 2020. Automated discovery of logical fallacies in legal argumentation. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 11.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Justus J Randolph. 2005. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. *Advances in Data Analysis and Classification*.

Bertrand Russell. 2013. *History of Western Philosophy: Collectors Edition*. Routledge.

Sougata Saha and Rohini Srihari. 2023. ArgU: A controllable factual argument generator. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8373–8388, Toronto, Canada. Association for Computational Linguistics.

Swarnadeep Saha, Prateek Yadav, Lisa Bauer, and Mohit Bansal. 2021. ExplaGraphs: An explanation graph generation task for structured commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7716–7740, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Aspect-controlled neural argument generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–396, Online. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Synthetic prompting: generating chain-of-thought demonstrations for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2020. Towards Controllable Biases in Language Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3239–3254, Online. Association for Computational Linguistics.

Zhivar Sourati, Filip Ilievski, Hông-Ân Sandlin, and Alain Mermoud. 2023. Case-based reasoning with language models for classification of logical fallacies. *arXiv preprint arXiv:2301.11879*.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Daniel Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Henning Wachsmuth, Nona Naderi, Ivan Habernal, Yufang Hou, Graeme Hirst, Iryna Gurevych, and Benno Stein. 2017. Argumentation quality assessment: Theory vs. practice. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 250–255, Vancouver, Canada. Association for Computational Linguistics.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation schemes*. Cambridge University Press.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation.

A Data Augmentation and Evaluation with LLMs

A.1 Generating Arguments with ChatGPT

Our prompt design for ChatGPT to generate fallacies follows a similar heuristic to (Liu et al., 2023), by introducing the task, defining the fallacy type it must generate, along with two examples of that particular fallacy type. Following the distribution in Fig. 3, we generate four fallacies for the same topic and feed the following prompt to ChatGPT to generate fallacies as negative preference data:

> You are given a topic *T*. Your task is to generate a {'supporting' or 'counter'} argument in the form of a *f-type*[a] logical fallacy in the context of the topic. It should not be longer than 25 words.
> *f-type* fallacy is defined as: {definition}
> examples of *f-type* are:
> {example 1}
> {example 2}
> Here is an example of *f-type* fallacy argument:
> {example of an argumentative fallacy}
> return {
> "topic": *T*, "fallacy": *f-type*, "argument": <...>
> }
>
> [a]Fallacy type that can be any of the thirteen types described in Table 9

Some examples of generated logical fallacies include: *"I know someone who smoked cannabis and became successful. Therefore, everyone who smokes cannabis will be successful.", "I know a few people who spend too much time on social media and have no real-life friends. Therefore, social media is terrible for society."*. Tab. 10 presents examples of samples in our preference dataset. Our augmented fallacy argument dataset consists of the train-test split in Tab. 2.

A.2 Prompting GPT-4 for Evaluation

To evaluate whether π_θ for a given method θ generates logical arguments compared to π_{SFT}, we use GPT-4 and evaluate the *win-rate* (e.g., how often does π_θ produce better arguments) by prompting GPT-4 with:

> Which of these arguments is better for the topic t and stance s:
> 1. $\pi_{\text{SFT}}(y|t,s)$
> 2. $\pi_\theta(y|t,s)$
>
> If both arguments are equally good, return 3 (Tie). The better argument is: <response>

We also evaluate how often models produce logical fallacies, which we call *fallacy-rate*, by prompting ChatGPT with:

> Consider the following *topic* t, stance s {supporting or counter} and argument $a = \pi_\theta(y|t,s)$:
> Topic: t
> Argument: a
> Out of all the following logical fallacy types {list of types from Tab. 9}
> would you qualify a as one of these logical fallacies? If not - return "None".
> If yes, which logical fallacy type is it? Let *f-type* be your answer. Return
> {"topic": t, "text": a, "fallacy type": *f-type* }

B Preference Optimization

Preference optimization is a crucial step in aligning language models to generate outputs that meet user preferences and objectives effectively. It involves the process of ensuring that the goals and preferences of AI systems align with those of their human users. To demonstrate models are capable of learning to distinguish logically sound text from logical fallacies, we assess the performance of four different preference optimization techniques, including PPO (Schulman et al., 2017), DPO (Rafailov et al., 2023), CPO (Xu et al., 2024) and KTO (Ethayarajh et al., 2024).

PPO. One widely used reinforcement learning optimization algorithm within RLHF is Proximal Policy Optimization (PPO). PPO (Schulman et al., 2017) is particularly favored due to its stability and efficiency. It iteratively updates the model's policy parameters by maximizing the expected cumulative reward while constraining the policy updates to a proximity threshold, preventing large policy changes that could destabilize learning.

DPO. More recently, Rafailov et al. (2023) introduced Direct Preference Optimization, which skips the reward modelling part that is necessary for PPO. DPO leverages an analytical mapping from reward functions to optimal policies, to transform a loss function over reward functions into a loss function over policies and avoids fitting a reward model, while still optimizing under given preferences.

CPO. Another recently introduced method is Contrastive Preference Optimization (CPO). Xu et al. (2024) introduced CPO as a derivation of DPO, to address some shortcomings of DPO, including memory and speed inefficiencies. (Xu et al., 2024) focuses mainly on machine translation, but the method can also be adapted to regular preference optimization for other tasks. They also incorporate a behaviour cloning regularizer to ensure that the policy does not deviate from the preferred data distribution.

KTO. Finally, the last method we evaluate is Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024). The authors introduce the concept of human-aware loss functions (HALOs), which implicitly model human biases and have been shown to perform better than non-HALOs. Their approach directly maximizes the utility of generations instead of maximizing preference likelihood, as is commonly done.

C Details of Experimental Setup

C.1 Hyperparameters

Training Parameters. For both base models, we train π_{SFT} and π_θ on 2 A100 GPUs for 3 epochs, using an Adam optimizer, with a learning rate of 2^{-4}. For the LoRA configuration, we select a rank $r = 16$, $\alpha = 32$ and a dropout of 5^{-2}. Regarding the specific training details of each method, we use $\beta = 0.25$ for DPO (Rafailov et al., 2023), which controls how much π_θ deviates from the reference model π_{SFT}. For both CPO (Xu et al., 2024) and KTO (Ethayarajh et al., 2024), we use $\beta = 0.1$, which controls the implicit reward. Regarding the reward model for PPO (Schulman et al., 2017), we use a binary logical fallacy classifier we trained using the chosen responses as non-fallacies and the rejected responses as fallacies. The classifier achieves accuracy and F1 over 95% in the detection of fallacy arguments, which makes us confident in using the model as a reward model for this particular task and we use the logits as rewards.

Decoding Parameters. At inference time, we generate arguments with the aligned models using the same decoding strategy. We use nucleus-sampling with $p = 0.75$ and top-k sampling with $k = 10$.

Hyperparameter Selection for FIPO. To optimize our custom loss function, defined in Equation 4, we conduct a series of experiments manipulating the hyperparameter λ (Equation 4) and the weights for the cross-entropy loss (Equation 3). Our initial step involved tuning the weights for the loss function. Given that our dataset consists of n pairs of preferred and dispreferred arguments as logical fallacies, it is crucial to differentiate different fallacy classes. It is thus more suitable to set the weights for each fallacy type as their frequency in the dataset, given by Equation 3 and the weight for the preferred responses as little as possible, which we set as the minimum of all the fallacy frequencies. We also evaluate different settings of λ, testing values of 0.1, 0.3, and 0.6. Our findings indicate that a higher λ effectively reduced the number of fallacies produced by the policies but adversely impacted the argument quality (*win-rate*). Conversely, a λ of 0.1 had minimal impact on improving the *fallacy-rate* . After assessing the trade-offs, we determined that a λ value of 0.3 provided the optimal balance between minimizing fallacies and maintaining a reasonable *win-rate* .

C.2 Context for Retrieval Augmented Generation (RAG)

We use the `wiki-dpr` database to add extra contextual information to prompts. RAG (Lewis et al., 2020) ensures that arguments are grounded in factual accuracy and contextually relevant information by retrieving relevant information from pre-existing knowledge sources. We retrieve relevant documents with the topics and add them to the prompts as context. For example, given the topic *Factory farming should not be banned*, the knowledge extracted includes *"The practice of dairy production in a factory farm environment has been criticized by animal welfare activists. The U.S. Food and Drug Administration states that no 'significant difference' has been found between milk from treated and non-treated cows. [...]"*

C.3 Licenses of Artifacts

The artifacts we employ, including datasets, packages, and models, are detailed in Tab. 12. Our usage of these artifacts is consistent with their intended purposes.

C.4 Detailed Setup of Human Annotations in Amazon Mechanical Turk

The privacy of these collected annotations for the pairwise comparison is under the university policy, and we use Amazon Mechanical Turk's services. The webpage we use for the comparative study

by human annotators is provided in Fig. 6. The workers are paid fairly for their annotating work.

D Additional Experimental Results

D.1 *Fallacy-Rate* in Zero-Shot Setting

Tab. 6 presents the distribution of fallacies detected with GPT-4 in the zero-shot setting using Llama-2 and Mistral.

Fallacy Types	Llama-2	Mistral	ChatGPT	Llama-2-RAG
Faulty Generalization	13	10	9	12
False Causality	3	4	4	5
Appeal To Emotion	2	1	1	2
Fallacy of Relevance	1	1	✗	3
False Dilemma	1	1	2	3
Circular Reasoning	29	21	3	12
Fallacy of Logic	6	✗	2	✗
Fallacy-Rate	55	38	21	37
Not A Fallacy	45	62	79	63

Table 6: *Fallacy-rate* according to GPT-4 on arguments generated by Llama-2, Mistral, ChatGPT, and a Llama-2 based RAG model. All of these models are in the zero-shot setting.

D.2 Results Regarding *Win-Rate*

Tab. 7 presents the results of win-rates computed by GPT-4 by comparing the policies with the corresponding SFT models and comparing which models win more often.

Win-Rate	$\pi_{\textbf{SFT}}$	π_θ	**Tie**
Llama-2 (7B)			
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{DPO}}$	35	61	4
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{PPO}}$	38	49	13
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{CPO}}$	43.5	52.5	4
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{KTO}}$	40.5	46	13.5
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{FIPO}}$	33	**63.5**	3.5
Mistral (7B)			
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{DPO}}$	49	48	3
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{PPO}}$	48	49	3
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{CPO}}$	46.75	51.5	1.75
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{KTO}}$	44.5	51.5	4
$\pi_{\textbf{SFT}}$ vs. $\pi_{\textbf{FIPO}}$	27	**68**	5

Table 7: *Win-rate* according to GPT-4. π_θ is the aligned policy, where θ designates the respective alignment method (e.g., when comparing $\pi_{\textbf{SFT}}$ with $\pi_{\textbf{DPO}}$, then the value 61 means $\pi_{\textbf{DPO}}$ wins 61% of the time.

Another interesting observation from Fig. 4 and Tab. 7 is the tie-rate. Annotating concise arguments to determine superiority is challenging for humans, often leading to the simpler selection of *Tie* when both arguments adequately address the topic. In fact, *Tie* was chosen more frequently in human evaluations compared to the automatic evaluation by GPT-4.

D.3 Examples of Fallacies Produced

Tab. 11 shows a few examples of arguments and fallacies produced by the models on a set of three topics.

E Human Annotations: Complementary Details

E.1 Using Humans to Classify Arguments

We use annotators to compute agreement scores with GPT-4's predictions, in order to justify the use of GPT-4 and its reliability. Another annotation task was conducted prior, and involved annotators tasked with classifying arguments into one of 13 fallacy types or as *Not a Fallacy*. Out Of 200 samples, each annotated by three different workers, a majority agreement was reached in only 47% of cases. This is likely due to the inherent difficulty of accurately identifying logical fallacies and leads to a loss of information if no agreement is reached. As such, we believe the easiest way to involve humans in computing the *fallacy-rate* is to task them with a binary decision instead.

E.2 Agreements with GPT-4: Randolph's κ

Randolph's κ is particularly well-suited for scenarios with rating imbalances (e.g., frequent "Agree" responses), as it does not assume an expected distribution of categories and instead reflects pure agreement. It avoids over-adjusting for chance in cases where one category dominates. Randolph's κ is computed as $\kappa = \frac{P_0 - P_e}{1 - P_e}$ where P_0 is the observed agreement:

$$P_0 = \frac{1}{Nn(n-1)} \sum_{i=1}^{N} (n_i^{agree})^2 + (n_i^{disagree})^2 - n \tag{5}$$

where N is the number of samples annotated, n is the number of annotators per sample (3 in our case) and $n_i^{(dis)agree}$ is the number of agreements and disagreements in each sample. $P_e = 0.5$, is the expected agreement for the two categories ('Agree' and 'Disagree').

F Ablation Study

To validate the effectiveness of our design, we conducted two ablation studies:

- Dataset Uniformity: The first study involves modifying the training dataset to include an equal number of samples for each fallacy type. Our original data exhibits a natural distribution where some fallacy types are more prevalent. For this study, we created a uniformly distributed dataset by downsampling, resulting in a dataset comprising 2,522 samples (194 per fallacy type).

- Unweighted Cross-Entropy: The second study examines the impact of applying FIPO with unweighted cross-entropy. Our design uses weighted cross-entropy based on fallacy-type frequency, penalizing models more for misclassifying common fallacies. This study uses unweighted cross-entropy to demonstrate the justification for our design, as the fallacy misclassification rates are higher with unweighted cross-entropy.

	Fallacy Rates
Current Design	17%
Uniformly Distributed Dataset over Fallacy-Types	37.5%
Unweighted Cross-Entropy	29%

Table 8: Ablation study proving the effectiveness of imbalanced fallacy types and weighted cross-entropy.

Term	Alternate Term (if any)	Definition
Faulty Generalization	N/A	When a conclusion is drawn about all or many instances of a phenomenon. It is an example of jumping to conclusions.
False Causality	N/A	Statement that jumps to a conclusion implying a causal relationship without supporting evidence.
Circular Reasoning	N/A	when the end of an argument comes back to the beginning without having proven itself.
Ad Populum	Appeal to Popularity	an argument based on affirming that something is real or better because the majority thinks so.
Ad Hominem	Personal Attack	Irrelevantly attack the person or some aspect of the person making the argument.
Fallacy of Logic	Logical Error	An error in the logical structure of an argument.
Appeal to Emotion	N/A	Manipulation of the recipient's emotions in order to win an argument
False Dilemma	Excluding Viable Alternatives	Presenting only two options or sides when there are many options or sides.
Equivocation	Use of Ambiguous Language	When a key term or phrase in an argument is used in an ambiguous way.
Fallacy of Extension	Exaggerating	Attacking an exaggerated or caricatured version of your opponent's position.
Fallacy of Relevance	Irrelevant Argument	Introducing premises or conclusions that have nothing to do with the subject matter.
Fallacy of Credibility	Attacking the Character of the Speaker	Attempts to disprove an argument by attacking the character of the speaker.
Intentional	Intentionally Wrong Argument	Some intentional (sometimes subconscious) action/-choice to incorrectly support an argument

Table 9: Definitions of different types of logical fallacies.

Prompt	Golden	Fallacy
Generate a supporting argument for the topic: Cannabis should be legal.	It's not a bad thing to make marijuana more available.	Why should we be worrying about legalizing cannabis when there are more important issues like poverty and hunger? (*Fallacy of Relevance*)
Generate a supporting argument for the topic: Urbanization is terrible for the planet.	Urbanization increases pollution.	Either we continue urbanization and destroy the planet, or we stop urbanization and hinder economic growth. (*False Dilemma*)
Generate a supporting argument for the topic: Research on embryonic stem cell should not be tax subsidized because for many it goes against their religious beliefs.	There are Christians who disagree with doing research on embryonic stem cells.	Those who support tax subsidies for embryonic stem cell research are godless and immoral. (*Ad Hominem*)

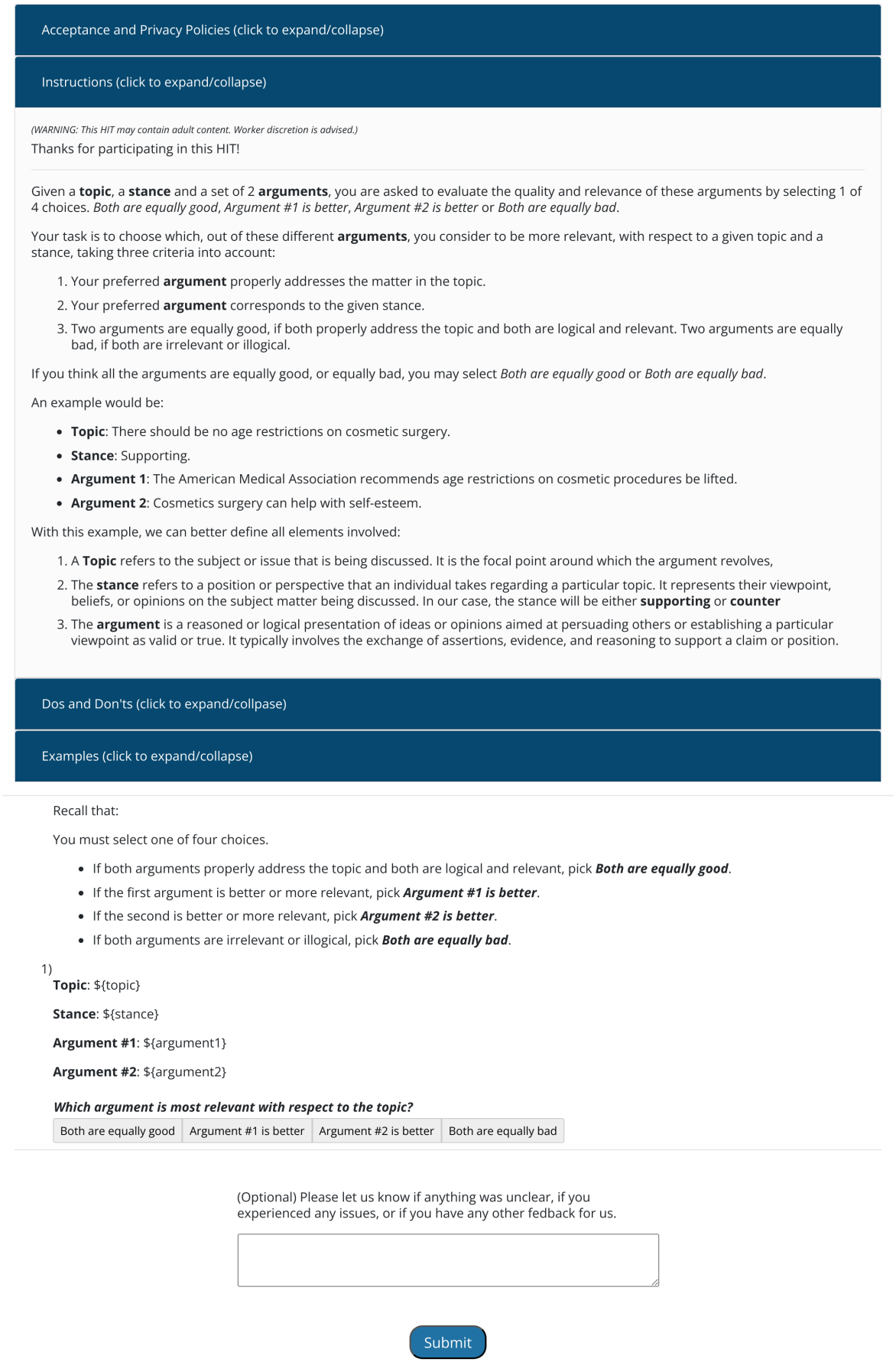
Table 10: Examples of samples from the preference dataset used for preference optimization. Golden arguments are retrieved from previous work (Saha et al., 2021), while fallacy arguments are generated using ChatGPT.

Topics	Surrogacy is an advantage for people. (Stance: Counter)	Executive are hard working. (Stance: Counter)	Companies should be allowed to decide workers' pay. (Stance: Counter)
SFT	Surrogacies are dangerous for children. (*Faulty Generalization*)	Executive do not work hard. (*Faulty Generalization*)	Compensation should not be based on what a company decides (*Not a Fallacy*)
DPO	Surplus of babies will cause more crime. (*Faulty Generalization*)	Executive do not care about the workers. (*Faulty Generalization*)	Pay should not be left to the company. It should reflect the cost of living. (*Not a Fallacy*)
KTO	Surrogacies are not good for everyone. (*Faulty Generalization*)	Executive have no time to work. (*Faulty Generalization*)	Compensation should not be decided by companies. (*Not a Fallacy*)
PPO	Surrogacies are not safe. (*Faulty Generalization*)	Executive can be lazy. (*Faulty Generalization*)	Compensation should not be decided by the company. (*Not a Fallacy*)
CPO	Surrogacy is a dangerous practice that puts women in harm's way. (*Faulty Generalization*)	Executive has access to money which makes them natural born abusers. (*Faulty Generalization*)	Anytime there is Imbalance of Power, the Countervailing power should regulate. (*Not a Fallacy*)
FIPO	Surrogates have no legal rights over their own bodies and this is not questioned. (*Not a Fallacy*)	Executive are lazy. (*Faulty Generalization*)	Workers should have some say in their pay; companies do not always make the best decisions. (*Not a Fallacy*)

Table 11: Examples of arguments generated by different preference optimization policies.

Artifacts	**Citation**	**Link**	**License**
LOGIC	(Jin et al., 2022)	https://github.com/causalNLP/logical-fallacy	N/A
EXPLAGRAPHS	(Saha et al., 2021)	https://explagraphs.github.io/	CC BY-SA 4.0
Debatepedia	(Cabrio and Villata, 2012)	N/A	N/A
PyTorch	(Paszke et al., 2019)	https://pytorch.org/	BSD-3 License
transformers	(Wolf et al., 2020)	https://huggingface.co/docs/transformers/index	Apache License 2.0
wandb	N/A	https://pypi.org/project/wandb/	MIT License
nltk	(Bird and Loper, 2004)	https://www.nltk.org/	Apache License 2.0
OpenAI API	N/A	https://platform.openai.com/docs/api-reference	MIT License
scikit-learn	(Pedregosa et al., 2011)	https://scikit-learn.org/stable/	BSD License

Table 12: Licenses of the artifacts utilized in this work, including datasets and major software packages, along with their respective licenses.



Acceptance and Privacy Policies (click to expand/collapse)

Instructions (click to expand/collapse)

(WARNING: This HIT may contain adult content. Worker discretion is advised.)

Thanks for participating in this HIT!

Given a **topic**, a **stance** and a set of 2 **arguments**, you are asked to evaluate the quality and relevance of these arguments by selecting 1 of 4 choices. *Both are equally good*, *Argument #1 is better*, *Argument #2 is better* or *Both are equally bad*.

Your task is to choose which, out of these different **arguments**, you consider to be more relevant, with respect to a given topic and a stance, taking three criteria into account:

1. Your preferred **argument** properly addresses the matter in the topic.
2. Your preferred **argument** corresponds to the given stance.
3. Two arguments are equally good, if both properly address the topic and both are logical and relevant. Two arguments are equally bad, if both are irrelevant or illogical.

If you think all the arguments are equally good, or equally bad, you may select *Both are equally good* or *Both are equally bad*.

An example would be:

- **Topic**: There should be no age restrictions on cosmetic surgery.
- **Stance**: Supporting.
- **Argument 1**: The American Medical Association recommends age restrictions on cosmetic procedures be lifted.
- **Argument 2**: Cosmetics surgery can help with self-esteem.

With this example, we can better define all elements involved:

1. A **Topic** refers to the subject or issue that is being discussed. It is the focal point around which the argument revolves,
2. The **stance** refers to a position or perspective that an individual takes regarding a particular topic. It represents their viewpoint, beliefs, or opinions on the subject matter being discussed. In our case, the stance will be either **supporting** or **counter**
3. The **argument** is a reasoned or logical presentation of ideas or opinions aimed at persuading others or establishing a particular viewpoint as valid or true. It typically involves the exchange of assertions, evidence, and reasoning to support a claim or position.

Dos and Don'ts (click to expand/collpase)

Examples (click to expand/collapse)

Recall that:

You must select one of four choices.

- If both arguments properly address the topic and both are logical and relevant, pick ***Both are equally good***.
- If the first argument is better or more relevant, pick ***Argument #1 is better***.
- If the second is better or more relevant, pick ***Argument #2 is better***.
- If both arguments are irrelevant or illogical, pick ***Both are equally bad***.

1)

Topic: ${topic}

Stance: ${stance}

Argument #1: ${argument1}

Argument #2: ${argument2}

Which argument is most relevant with respect to the topic?

Both are equally good	Argument #1 is better	Argument #2 is better	Both are equally bad

(Optional) Please let us know if anything was unclear, if you experienced any issues, or if you have any other fedback for us.

Submit

Figure 6: User interface used for human annotation on Amazon Mechanical Turk, where annotators compare pairs of arguments generated by different models to determine the superior argument.

G Fallacy-Informed Preference Optimization: Loss Definition

In our work, FIPO combines the CPO loss (Xu et al., 2024) for preference optimization with our classification loss. We selected CPO because it achieves the optimal balance between *win-rate* and *fallacy-rate* (Fig. 4, Tab. 4). Moreover, the CPO approach is designed to make the model learn rejections more accurately, which fits our study.

CPO Loss

CPO is a reference-free preference optimization method. It extends DPO (Rafailov et al., 2023), modifying the DPO loss by using a Uniform model instead of a reference model. The DPO loss is defined as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]$$

The CPO loss first approximates the DPO loss using a uniform reference model:

$$\mathcal{L}(\pi_\theta, U) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\pi_\theta(y_w|x) - \beta\log\pi_\theta(y_l|x)\Big)\Big]$$

and defines the CPO loss as $\mathcal{L}_{\text{CPO}}$:

$$\mathcal{L}_{\text{CPO}}(\pi_\theta) = \min_\theta\Big(\mathcal{L}(\pi_\theta, U) - \mathbb{E}_{(x,y_w)\sim\mathcal{D}}[\log\pi_\theta(y_w|x)]\Big) \tag{6}$$

which combines the preference loss with a negative log-likelihood (NLL) term. The $\min$ term ensures π_θ does not deviate from the preferred data distribution.

FIPO Loss

First, recall our preference dataset $\mathcal{D} = \{t^{(i)}, s^{(i)}, y_w^{(i)}, y_l^{(i)}, k^{(i)}\}$, with t the topic, s the stance, y_w the preferred valid argument, y_l the dispreferred fallacious argument and k the fallacy type of y_l.
Now recall the definitions for our classification loss in Section 4.4. We define the probability p_k of each fallacy type k as the output of the linear layer after we feed it the last hidden state after the forward pass into the model π_θ:

$$p_k(h_\theta^L(y|t,s)) = \text{Softmax}(Wh_\theta^L(y|t,s) + b)_k; \text{ with Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \tag{7}$$

where W is the linear layer's weight matrix, b is the corresponding bias term and $h_\theta^L(y|t,s)$ is the last layer's hidden state. Additionally, we use weights in our classification loss defined as the frequency of the fallacy types in our dataset.

$$w_k = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\mathbb{1}_{k^{(i)}=k} \qquad \text{For } \textit{fallacy-types } k \tag{8}$$

$$w_0 = \min_k w_k \qquad \text{For the class } \textit{'Not a Fallacy'} \text{ (class 0)} \tag{9}$$

Our method introduces a weighted cross-entropy loss which we use in addition to the preference loss:

$$\mathcal{L}_{\text{CLF}} = -\mathbb{E}_{(t,s,y_w,y_l,k)\sim\mathcal{D}}\Big[w_0\log p_0(h_\theta^L(y_w|t,s)) + w_k\log p_k(h_\theta^L(y_l|t,s))\Big] \tag{10}$$

FIPO is thus reference-free and the loss $\mathcal{L}_{\text{FIPO}}$ is then defined with the following term:

$$\begin{aligned}\mathcal{L}_{\text{FIPO}} = \mathcal{L}_{\text{CPO}} + \lambda\mathcal{L}_{\text{CLF}} = -\mathbb{E}_{(t,s,y_w,y_l,k)\sim\mathcal{D}} \\ \Big[\underbrace{\min_\theta\log\sigma\Big(\beta\log\pi_\theta(y_w|t,s) - \beta\log\pi_\theta(y_l|t,s)\Big) + \log\pi_\theta(y_w|t,s)}_{\mathcal{L}_{\text{CPO term}}} \\ + \lambda\underbrace{\Big(w_0\log p_0(h_\theta^L(y_w|t,s)) + w_k\log p_k(h_\theta^L(y_l|t,s)\Big)}_{\mathcal{L}_{\text{CLF term}}}\Big]\end{aligned}$$