ANCHORED ALIGNMENT FOR SELF-EXPLANATIONS ENHANCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we introduce a methodology for alignment designed to enhance the ability of large language models (LLMs) to articulate their reasoning—*self-explanation*—even in the absence of annotated rationale explanations. Our alignment methodology comprises three key components: explanation quality assessment, self-instruction dataset generation, and model alignment. Additionally, we present a novel technique called *Alignment with Anchor Preference Pairs*, which improves the selection of preference pairs by categorizing model outputs into three groups: consistently correct, consistently incorrect, and variable. By applying tailored strategies to each category, we enhance the effectiveness of Direct Preference Optimization (DPO). Our experimental results demonstrate that this approach significantly improves explanation quality while maintaining accuracy compared to other fine-tuning strategies.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable capabilities across various tasks. However, fine-tuning these models for specific applications often leads to a critical trade-off: improvements in one area may compromise the model's generalization capabilities (Yang et al., 2024; Kirk et al., 2024). In our study, we aim to enhance a secondary task—specifically, the model's ability to articulate reasoning processes in natural language, a skill known as *self-explanation* (Madsen et al., 2024a)—in parallel with the primary task, despite the constraint of not having human-annotated rationales.

The lack of annotated data of both high- and low-quality explanations can be framed in the context of model aligning without human preference data. Recent research has explored ways to align LLMs without direct human input. Some approaches generate self-instruct data to fine-tune models (Wang et al., 2023; Chen et al., 2023; Gulcehre et al., 2023), while others, like Bai et al. (2022); Yuan et al. (2024); Wu et al. (2024), use LLM-generated feedback to train reward models. Building on these advancements, we propose an end-to-end approach to align LLMs on classification tasks while also ensuring the generation of high-quality self-explanations, even without annotated data for this secondary task. Our approach integrates three core components: evaluating generated explanations, creating self-instruct datasets, and aligning the model. Additionally, we introduce *Alignment with Anchor Preference Pairs*, a method that improves preference pair selection by categorizing model responses into three groups: consistently correct, consistently incorrect, and variable. For each category, we apply tailored strategies to construct preference pairs, which are then used in the Direct Preference Optimization (DPO) phase (Rafailov et al., 2023). Our results demonstrate that this method consistently improves explanation quality, mitigating the degradation caused by SFT. Moreover, we show that using anchor preference pairs outperforms self-alignment strategies that rely solely on judge-based evaluations for preference pair selection.

Our contributions are summarized as follows:

(i). We introduce a framework for the qualitative assessment of self-explanations, designed to evaluate how effectively the model conveys its reasoning.

(ii). We analyze how supervised fine-tuning for classification tasks affects the quality of self-explanations. Our findings demonstrate that while SFT improves classification accuracy, it often reduces explanation quality, underscoring the need for improved alignment strategies.

(iii). We propose a novel method, *Alignment with Anchor Preference Pairs*, for constructing high-quality preference pairs when building self-instruct datasets. This method uses the model's behavior on each input prompt to apply specific strategies while creating preference pairs. Our approach consistently outperforms other methods that rely solely on judge-based evaluations for selecting preference pairs.

(iv). We develop an end-to-end methodology for aligning LLMs to downstream classification tasks while maintaining the quality of their self-explanations, even in the absence of explanation-rich datasets.

2 A Framework for Qualitative Assessment of Self-Explanations

2.1 Quality Criteria for Effective Self-Explanations

To assess self-explanation quality, we focus on the model's ability to effectively communicate its reasoning. This approach differs from previous work that emphasized trustworthiness metrics such as faithfulness (Madsen et al., 2024b;a; Lanham et al., 2023; Lyu et al., 2023; Turpin et al., 2023; Parcalabescu & Frank, 2024) and truthfulness (Zhang et al., 2024; Sharma et al., 2023; Burns et al., 2022; Joshi et al., 2024). We evaluate self-explanations based on the following criteria:

1. **Logical coherence**: The explanation should follow a clear and logical reasoning process, with all components cohesively connected to form a unified, non-contradictory narrative.
2. **Clarity**: The explanation must present ideas clearly and precisely, using appropriate terminology to effectively communicate complex concepts without unnecessary complexity.
3. **Relevance**: The explanation should comprehensively address the task at hand, directly answering the specific context or requirements without omitting critical information.
4. **Depth of argumentation**: The explanation must provide strong reasoning and credible evidence to support its conclusions, reflecting a deep understanding of the task.
5. **Factual accuracy**: This criterion assesses the correctness of individual claims within the explanation. While related to truthfulness, factual accuracy focuses on whether specific statements align with established knowledge.

2.2 Self-Explanations Evaluation Methodology

Let $\mathcal{M}$ represent a large language model tasked with generating responses for a classification problem. Each response consists of two components: a self-explanation, denoted as ε_i, and a predicted classification label, $\hat{y}_i$, corresponding to an input prompt x_i. The self-explanation ε_i is produced by prompting the model to articulate its reasoning before providing a final prediction, following the Chain-of-Thought prompting strategy (Wei et al., 2022).

Our methodology is inspired by recent approaches that utilize LLMs as evaluators of other models' outputs (Dubois et al., 2023; Li et al., 2024; Fernandes et al., 2023; Bai et al., 2023; Saha et al., 2024). This approach has shown versatility, extending beyond simple evaluation to various applications in model improvement and self-alignment strategies. For instance, researchers have employed this framework to generate self-instruct data for fine-tuning models (Wang et al., 2023; Chen et al., 2023; Gulcehre et al., 2023) and to create feedback for training reward models (Bai et al., 2022; Yuan et al., 2024; Wu et al., 2024).

For our evaluation, we employ a more capable model, $\mathcal{M}_{\text{Judge}}$, to assess the quality of self-explanations ε_i based on predefined criteria (detailed in Section 2.1). The evaluation process proceeds as follows:

1. For each criterion κ, $\mathcal{M}_{\text{Judge}}$ assigns a qualitative verdict $v_{i,\kappa}$ from the set {excellent, satisfactory, needs improvement, unsatisfactory}. The prompt used by $\mathcal{M}_{\text{Judge}}$ is provided in Appendix C.2.
2. Each verdict $v_{i,\kappa}$ is mapped to a numerical score $s_{i,\kappa}$ (see Appendix C.1).
3. The overall score for an explanation, s_i, is computed as the sum of scores across all criteria: $s_i = \sum_{k=1}^{K} s_{i,k}$

2.3 Pairwise Model Evaluation

To assess the quality of self-explanations generated by different models, we adopt a pairwise evaluation strategy consistent with previous work (Chen et al., 2023; Yuan et al., 2024; Wu et al., 2024). For each input prompt x_i, we generate N sample self-explanations, capturing the inherent variability in model outputs. The score for the n-th response is denoted as s_i^n, with the corresponding explanation and prediction represented by the pair $(\varepsilon_i^n, \hat{y}_i^n)$.

For a given prompt x_i, we conduct N^2 pairwise comparisons between the explanations generated by two models, $\mathcal{M}_1$ and $\mathcal{M}_2$. A *win* for model $\mathcal{M}_1$ is defined when:

$$s_i^n(\mathcal{M}_1) > s_i^m(\mathcal{M}_2)$$

where $n, m \in \{1, \ldots, N\}$. The overall win rate $W(\mathcal{M}_1, \mathcal{M}_2)$ is then calculated as follows:

$$W(\mathcal{M}_1, \mathcal{M}_2) = \frac{1}{|\mathcal{X}|} \sum_{x_i \in \mathcal{X}} \left(\frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \mathbb{1}[s_i^n(\mathcal{M}_1) > s_i^m(\mathcal{M}_2)] \right)$$

Here, $\mathcal{X}$ denotes the set of all prompts, while $\mathbb{1}[\cdot]$ represents the indicator function that returns 1 if the condition is true and 0 otherwise. This approach facilitates a nuanced comparison of model performance by taking into account the distribution of explanation qualities, rather than relying solely on single-point estimates. The rates for ties, defined as $s_i^n(\mathcal{M}_1) = s_i^m(\mathcal{M}_2)$, and losses, defined as $s_i^n(\mathcal{M}_1) < s_i^m(\mathcal{M}_2)$, are computed in a similar manner (see Appendix A). Throughout the evaluations presented in this work, $\mathcal{M}_2$ refers to the baseline model $\mathcal{M}_{\text{base}}$.

3 Self-Explanation Alignment with Anchor Preference Pairs

In this section, we introduce a methodology for alignment designed to enhance the ability of large language models (LLMs) to articulate their reasoning—*self-explanation*—even in the absence of annotated rationale explanations. However, we assume access to human-annotated data in the form of classification datasets for domain-specific adaptation, reflecting a common constraint in real-world applications, where comprehensive explanation data is often scarce or prohibitively expensive compared to classification datasets.

Building on prior work (Bai et al., 2022; Wang et al., 2023; Yuan et al., 2024; Wu et al., 2024), our alignment methodology incorporates familiar components such as self-instruction dataset generation, human-free evaluation of candidate responses using *LLM-as-Judge*, preference pair selection, and model alignment.

However, our approach differs from previous methods in two key ways: First, for the assessment of candidate responses, we use the evaluation explanation quality framework introduced in Sections 2.1 and 2.2. Second, we propose a novel technique, *Alignment with Anchor Preference Pairs*, which improves preference pair selection by categorizing model outputs into three groups: consistently correct, consistently incorrect, and variable. By applying tailored strategies to each category, we enhance the effectiveness of DPO.

The steps of the methodology are as follows:

1. Supervised fine-tuning of the base model $\mathcal{M}_{\text{Base}}$ specifically on a target classification task, resulting in $\mathcal{M}_{\text{SFT}}$.
2. Instruct $\mathcal{M}_{\text{SFT}}$ to generate multiple explanation-prediction pairs for each prompt, and evaluate the quality of these self-explanations using the methodology outlined in Sections 2.1 and 2.2. During alignment, the base model $\mathcal{M}_{\text{Base}}$ acts as the judge $\mathcal{M}_{\text{Judge}}$, ensuring the process remains self-contained.
3. Construct an alignment dataset by selecting preference pairs using an anchor-based strategy (see Section 3.3).
4. Align $\mathcal{M}_{\text{SFT}}$ via DPO with the dataset created in the third step, producing the aligned model $\mathcal{M}_{\text{Anchor}}$.

3.1 Supervised Fine-Tuning without Annotated Explanations

We fine-tuned the base model, $\mathcal{M}_{\text{Base}}$, on classification datasets (the primary task) to obtain $\mathcal{M}_{\text{SFT}}$, simulating scenarios where explanation annotations are unavailable. To replicate typical domain-specific adaptations and avoid potential gains from multi-task learning, we fine-tuned a separate model for each task. During fine-tuning, loss was calculated only on the target tokens corresponding to the correct choice sentence, excluding the system instruction and question. We generated the full text of the selected option to provide richer context and preserve the model's text generation capabilities. Details on datasets and training setups are provided in Section 4.1.

3.2 Self-Instruction Creation

We generate self-instruct data for alignment as follows:

1. **Generate candidate responses:** We sample N diverse pairs of explanations and predictions from $\mathcal{M}_{\text{SFT}}$, denoted as $\{\varepsilon_i^n, \hat{y}_i^n\}_{n=1}^N$, where ε_i^n represents the explanation for the n-th prediction $\hat{y}_i^n$ corresponding to the prompt x_i.
2. **Evaluate candidate responses:** We use the methodology described in Section 2.2 to evaluate the self-explanations generated from the candidate responses, assigning a score s_i^n to each explanation ε_i^n. During the creation of the self-instruct dataset, we employ $\mathcal{M}_{\text{base}}$ as the judge ($\mathcal{M}_{\text{Judge}}$). This ensures that the model alignment process remains self-contained, without the need for external models, except for evaluation purposes.

3.3 Preference Pairs via Anchor Selection

We introduce a method to enhance the selection of preference pairs by categorizing model responses into three groups: consistently correct, consistently incorrect, and variable. For each category, we apply specific strategies to construct preference pairs, which are then used in during the DPO phase. To evaluate the model's consistency on a given input prompt, a ground truth reference, or *anchor*, is required. We use a classification task as the probing mechanism.

Preference Pairs for Consistently Correct Prompts: For input prompts x_i where $\mathcal{M}_{\text{SFT}}$ consistently produces correct answers (i.e., $\hat{y}_i^n = y_i$ for all $n \in \{1, \ldots, N\}$), preference pairs are constructed based on the quality of the explanations. Let s_i^n denote the score assigned by the judge $\mathcal{M}_{\text{Judge}}$ to the n-th explanation ε_i^n for prompt x_i. We define two sets: $\mathbb{A}_i^w = \{\varepsilon_i^n : s_i^n = \max_{j\in\{1,\ldots,N\}} s_i^j\}$, which contains all explanations that achieve the highest score for prompt x_i, and $\mathbb{A}_i^l = \{\varepsilon_i^n : s_i^n < \max_{j\in\{1,\ldots,N\}} s_i^j\}$, which includes all explanations with scores lower than the maximum for prompt x_i.

Preference Pairs for Variable Performance: For input prompts x_i where $\mathcal{M}_{\text{SFT}}$ produces a mix of correct and incorrect predictions (i.e., $\hat{y}_i^n \neq y_i$ for some $n \in \{1, \ldots, N\}$), preference pairs are constructed contrastively. We define the set $\mathbb{B}_i^w = \{\varepsilon_i^n : \hat{y}_i^n = y_i\}$, which contains explanations associated with correct predictions. From this set, we extract $\mathbb{A}_i^w \subseteq \mathbb{B}_i^w$, the subset of explanations with the highest scores assigned by $\mathcal{M}_{\text{Judge}}$, i.e., $\mathbb{A}_i^w = \{\varepsilon_i^n \in \mathbb{B}_i^w : s_i^n = \max_{j\in\mathbb{B}_i^w} s_i^j\}$. The set $\mathbb{A}_i^l = \{\varepsilon_i^n : \hat{y}_i^n \neq y_i$ and $s_i^n < \max_{j\in\mathbb{A}_i^w} s_i^j\}$ contains explanations corresponding to incorrect predictions, with scores lower than the maximum score in $\mathbb{A}_i^w$.

Preference Pairs for Consistently Incorrect Prompts: For prompts where all predictions from $\mathcal{M}_{\text{SFT}}$ are incorrect (i.e., $\hat{y}_i^n \neq y_i$ for all $n \in \{1, \ldots, N\}$), all corresponding explanations are placed in the set $\mathbb{A}_i^l$. To generate a winning explanation, we employ the $\mathcal{M}_{\text{Base}}$ model in a consultant role, similar to the LLM-as-a-Debater approach proposed by Khan et al. (2024). Since the inference hyperparameters for the LLM in this consulting role might differ from those used during the generation of preference pairs, we refer to this model as $\mathcal{M}_{\text{Debater}}$ to avoid confusion. Specifically, we provide the correct answer y_i to the LLM and request an argument supporting this answer, which is then assigned to the set $\mathbb{A}_i^w$ as the winning explanation.

Finally, preference pairs are constructed for each instruction prompt x_i by randomly sampling ε_i^w from $\mathbb{A}_i^w$ as the winning explanation and ε_i^l from $\mathbb{A}_i^l$ as the losing explanation. The resulting preference pair is denoted as $(x_i, \varepsilon_i^w, \varepsilon_i^l)$. The detailed algorithm is presented in Algorithm 1.

Algorithm 1 Generating Preference Pairs Via Anchor Selection

```
Input: Instruction prompt $x_i$, model predictions $\{\hat{y}_i^n\}_{n=1}^N$, true label $y_i$, judge model $\mathcal{M}_{\text{Judge}}$, debater model $\mathcal{M}_{\text{Debater}}$
Output: Preference pairs $(x_i, \varepsilon_i^w, \varepsilon_i^l)$
Initialize: $\mathbb{A}_i^w \leftarrow \emptyset$, $\mathbb{A}_i^l \leftarrow \emptyset$
if $\hat{y}_i^n = y_i$ for all $n \in \{1, \dots, N\}$ then                ▷ Consistently Correct Prompts
    for each explanation $\varepsilon_i^n$ do
        Compute score $s_i^n$ from $\mathcal{M}_{\text{Judge}}$
    end for
    $\mathbb{A}_i^w \leftarrow \{\varepsilon_i^n : s_i^n = \max_{j \in \{1,\dots,N\}} s_i^j\}$
    $\mathbb{A}_i^l \leftarrow \{\varepsilon_i^n : s_i^n = \min_{j \in \{1,\dots,N\}} s_i^j\}$
else if $\hat{y}_i^n \neq y_i$ for some $n \in \{1, \dots, N\}$ then     ▷ Variable Performance Prompts
    $\mathbb{B}_i^w \leftarrow \{\varepsilon_i^n : \hat{y}_i^n = y_i\}$
    $\mathbb{A}_i^w \leftarrow \{\varepsilon_i^n \in \mathbb{B}_i^w : s_i^n = \max_{j \in \mathbb{B}_i^w} s_i^j\}$
    $\mathbb{A}_i^l \leftarrow \{\varepsilon_i^n : \hat{y}_i^n \neq y_i \wedge s_i^n < \max_{j \in \mathbb{A}_i^w} s_i^j\}$
else                                                                     ▷ Consistently Incorrect Prompts
    $\mathbb{A}_i^l \leftarrow \{\varepsilon_i^n : \hat{y}_i^n \neq y_i$ for all $n \in \{1, \dots, N\}\}$
    Generate argument $\varepsilon_i^{\text{debater}}$ using $\mathcal{M}_{\text{Debater}}$ given $y_i$
    $\mathbb{A}_i^w \leftarrow \{\varepsilon_i^{\text{debater}}\}$
end if
Sample $\varepsilon_i^w$ from $\mathbb{A}_i^w$
Sample $\varepsilon_i^l$ from $\mathbb{A}_i^l$
Return $(x_i, \varepsilon_i^w, \varepsilon_i^l)$
```

4 Experiments

In all experiments, we utilized `Llama-3-8B-Instruct` as our base model. Our study involved comparing four distinct model configurations:

1. $\mathcal{M}_{\text{Base}}$: The base model, which remains unmodified.
2. $\mathcal{M}_{\text{SFT}}$: This model was obtained by performing supervised fine-tuning on the $\mathcal{M}_{\text{Base}}$ model using only classification tasks, simulating scenarios where explanation annotations are not available.
3. $\mathcal{M}_{\text{Rank}}$: The $\mathcal{M}_{\text{SFT}}$ model was further refined using DPO, employing a self-instruct dataset constructed from rank-ordered preference pairs derived solely from judge-based evaluations of the explanations. This approach aligns with methodologies described in Bai et al. (2022); Wang et al. (2023); Yuan et al. (2024); Wu et al. (2024).
4. $\mathcal{M}_{\text{Anchor}}$ (ours): Similar to $\mathcal{M}_{\text{Rank}}$, this model was refined using DPO but utilized a self-instruct dataset created with our proposed anchored method for selecting preference pairs.

Since both $\mathcal{M}_{\text{Rank}}$ and $\mathcal{M}_{\text{Anchor}}$ undergo an additional stage of DPO alignment with the self-instruct dataset, we will collectively refer to them as self-aligned models in comparison to $\mathcal{M}_{\text{SFT}}$.

4.1 Experimental Setup

Datasets: We selected four datasets for our experiments: `AQuA-Rat` (Ling et al., 2017), `ARC-Challenge` (Clark et al., 2018), `LogiQA` (Liu et al., 2020), and `OpenbookQA` (Mihaylov et al., 2018). These datasets are established benchmarks for reasoning tasks, requiring a challenging reasoning process, which makes them an ideal fit for evaluating the quality of self-explanations. A key factor in their selection was the size of their training sets, which provided a sufficient number of input prompts to support the creation of the self-instruction dataset. In the case of `AQuA-Rat`, we sampled 5,000 examples due to computational constraints. For evaluation, we used the test split of each dataset.

SFT Training Details: For $\mathcal{M}_{\text{SFT}}$, we used the AdamW optimizer with a learning rate of 5×10^{-5} for one epoch, following a cosine schedule with 10% warmup steps. Gradient clipping was set to 0.3, and we used an effective batch size of 12. Loss was computed only on the assistant's completions. Instead of fine-tuning the entire model, we applied a LoRA adapter ($\alpha = 128$, dropout = 0.05, rank $r = 256$) to all linear layers. LoRA adapters were used to accelerate training and to act as a regularization method (Biderman et al., 2024), addressing the overfitting tendencies of DPO (Thakkar et al., 2024), which is applied during the later alignment phase.

Self-Instruct Dataset: To ensure the integrity of our evaluation process, we created separate self-instruct datasets for each benchmark. These datasets were built using input prompts specific to each task, ensuring that the DPO alignment data remained uncontaminated by exposure to multiple tasks. This approach prevents the artificial inflation of results that could occur if models were aligned across diverse tasks, unlike SFT models, which are trained on one classification task at a time. To create the self-instruct dataset for aligning $\mathcal{M}_{\text{Rank}}$ and $\mathcal{M}_{\text{Anchor}}$, we sampled $N = 4$ responses from $\mathcal{M}_{\text{SFT}}$ for each input prompt (settings: temperature $T = 0.6$ and top-k value of 0.9). This sample size provided a reasonable assessment of the model's consistency and variability. The prompt used is presented in Appendix D. In cases of consistently incorrect responses, $\mathcal{M}_{\text{Base}}$ was employed as the debater model (`Llama-3-8B-Instruct`) with adjusted parameters ($T = 0.5$, top-k 0.9). The responses were scored by $\mathcal{M}_{\text{Judge}}$, which was the same base model, ensuring a self-contained alignment process. This setup differs from the evaluation phase, where a more capable model serves as the judge. The scoring of explanations followed the methodology outlined in Section 2.2, with $\mathcal{M}_{\text{Judge}}$ using fixed inference parameters ($T = 0$). For $\mathcal{M}_{\text{Rank}}$, preference pairs were chosen based on the assigned scores, with the highest-scoring explanation designated as the winner, and the losing explanation randomly selected from the remaining candidates. The preference pairs for $\mathcal{M}_{\text{Anchor}}$ were selected using the methodology outlined in Section 3.3, and these pairs were then used to align the models through DPO.

DPO Training Details: For DPO-aligned models ($\mathcal{M}_{\text{Rank}}$, $\mathcal{M}_{\text{Anchor}}$), we used similar hyperparameters as in the SFT phase but reduced the learning rate to 5×10^{-7} and trained for 2.6k steps with an effective batch size of 6. The DPO process used a β value of 0.1 and updated the LoRA weights obtained during SFT.

Evaluation: We evaluated our models along two main dimensions: prediction accuracy and self-explanation quality. To account for variability in model outputs, we generated $N = 16$ explanation-prediction pairs per input prompt. The inference settings mirrored those used to create the self-instruction dataset, with a temperature of $T = 0.6$ and top-k set to 0.9 and the same prompt used during the creation of the self-instruct dataset (see Appendix D.3). Average prediction accuracy was used to measure performance on downstream tasks. To assess self-explanation quality, we performed head-to-head comparisons between the aligned models ($\mathcal{M}_{\text{SFT}}$, $\mathcal{M}_{\text{Rank}}$, $\mathcal{M}_{\text{Anchor}}$) and the base model ($\mathcal{M}_{\text{Base}}$). These comparisons followed the methodology outlined in Section 2.2, employing `Llama-3-70B-Instruct` as $\mathcal{M}_{\text{Judge}}$.

4.2 Results

Table 1 reports the average classification accuracy for each model, along with pairwise comparisons of self-explanation quality across multiple benchmark datasets. The win, tie, and loss rates are calculated by comparing the aligned models against $\mathcal{M}_{\text{Base}}$.

4.2.1 Impact of Supervised Fine-Tuning on Self-Explanations

We observed a significant trade-off in evaluation results before and after applying supervised fine-tuning on a classification task (see Table 1). While SFT notably improved classification accuracy, it resulted in a substantial decline in the quality of self-explanations compared to the base model. The decline in explanation quality, as measured by the win-loss rate difference (Δ_{W-L}), ranged from 15.6% to 30.9% across benchmarks.

Building on evidence that supervised fine-tuning can improve performance on specific tasks at the expense of a model's generalization abilities (Yang et al., 2024; Kirk et al., 2024), we hypothesize that this decline occurs because the task of selecting predefined answers does not inherently encourage the model to articulate its reasoning, leading to a specialization that diminishes the quality of the generated explanations.

Table 1: **Comparison of Aligned Models**. Average accuracy is presented along with head-to-head comparisons for self-explanation quality. The results show that $\mathcal{M}_{\text{Anchor}}$ achieves comparable or superior accuracy and attains the highest win rate when compared to $\mathcal{M}_{\text{SFT}}$ and $\mathcal{M}_{\text{Rank}}$.

Dataset	$\mathcal{M}_{\textbf{Base}}$ **Acc. (%)**	$\mathcal{M}_{\textbf{Align}}$ **Type**	$\mathcal{M}_{\textbf{Align}}$ **Acc. (%)**	ε **Win Rate Eval. (%)**			Δ_{W-L}
				Win ↑	**Tie**	**Loss** ↓	
AQuA Rat	$47.1_{\pm 2.9}$	$\mathcal{M}_{\text{SFT}}$	$47.7_{\pm 2.7}$	11.3	61.9	26.9	-15.6
		$\mathcal{M}_{\text{Rank}}$	$48.3_{\pm 2.1}$	11.5	63.1	25.4	-13.9
		$\mathcal{M}_{\text{Anchor}}$	$51.1_{\pm 3.0}$	12.6	70.8	16.7	-4.1
ARC-Challenge	$76.4_{\pm 0.7}$	$\mathcal{M}_{\text{SFT}}$	$81.0_{\pm 0.7}$	9.6	54.4	36.0	-26.4
		$\mathcal{M}_{\text{Rank}}$	$81.9_{\pm 1.1}$	17.72	61.4	20.9	-3.18
		$\mathcal{M}_{\text{Anchor}}$	$82.0_{\pm 0.9}$	21.5	60.3	18.2	3.3
LogiQA	$41.4_{\pm 1.1}$	$\mathcal{M}_{\text{SFT}}$	$45.2_{\pm 0.7}$	14.7	39.8	45.6	-30.9
		$\mathcal{M}_{\text{Rank}}$	$46.0_{\pm 1.5}$	22.0	50.1	27.9	-5.9
		$\mathcal{M}_{\text{Anchor}}$	$46.6_{\pm 2.2}$	26.6	53.8	19.7	6.9
OpenbookQA	$71.7_{\pm 1.3}$	$\mathcal{M}_{\text{SFT}}$	$87.4_{\pm 1.1}$	11.3	54.0	34.6	-23.3
		$\mathcal{M}_{\text{Rank}}$	$87.0_{\pm 1.1}$	15.4	60.1	24.5	-9.1
		$\mathcal{M}_{\text{Anchor}}$	$87.0_{\pm 0.9}$	16.7	59.6	23.7	-7

These findings highlight the necessity for alignment techniques that can preserve high-quality explanations in situations where datasets with annotated explanations are unavailable for fine-tuning.

4.2.2 Analysis of Self-Aligned Models

Prediction Accuracy: Our experiments demonstrate that the self-aligned models, $\mathcal{M}_{\text{Rank}}$ and $\mathcal{M}_{\text{Anchor}}$, maintain the classification accuracy improvements achieved by the seed model, $\mathcal{M}_{\text{SFT}}$, over the base model, $\mathcal{M}_{\text{Base}}$ (see Table 1). Notably, $\mathcal{M}_{\text{Anchor}}$ consistently achieves the highest, or at least comparable, accuracy across all datasets. We believe that this improvement can be attributed to the fact that the model's predictions ($\hat{y}_i$) are compared to the ground truth (y_i) to determine how to treat the self-explanation (e_i) during the selection of preference pairs while employing the anchor strategy, thereby providing a more informative learning signal.

Self-Explanation Quality: Pairwise evaluations of self-explanation quality (see Table 1) indicate that the initial decline in explanation performance observed in $\mathcal{M}_{\text{SFT}}$ is partially inherited by both $\mathcal{M}_{\text{Rank}}$ and $\mathcal{M}_{\text{Anchor}}$, as they use $\mathcal{M}_{\text{SFT}}$ as the seed model during the DPO alignment phase. Nevertheless, both $\mathcal{M}_{\text{Rank}}$ and $\mathcal{M}_{\text{Anchor}}$ demonstrate significant improvements in explanation quality compared to $\mathcal{M}_{\text{SFT}}$, with $\mathcal{M}_{\text{Anchor}}$ exhibiting the strongest performance. Across all datasets, $\mathcal{M}_{\text{Anchor}}$ consistently achieves the highest win rates and lowest loss rates. Compared to the base model, $\mathcal{M}_{\text{Anchor}}$ also shows positive Δ_{W-L} margins on ARC-Challenge (3.3%) and LogiQA (6.9%), indicating that its explanations are more frequently preferred. Additionally, it significantly narrows the gap in explanation quality introduced by SFT across the remaining benchmark datasets.

4.2.3 Analysis of Individual Evaluation Dimensions

Figure 1 presents the average scores for each evaluation criterion used to assess self-explanations, as described in Section 2.1, for all evaluated models across the benchmark datasets.

Overall, the self-aligned models outperform $\mathcal{M}_{\text{SFT}}$ across all evaluation criteria, with $\mathcal{M}_{\text{Anchor}}$ consistently achieving better results than $\mathcal{M}_{\text{Rank}}$.

Additionally, we observe that the degradation in self-explanation quality due to SFT varies significantly depending on the dataset used for fine-tuning. Two notable trends emerge from the analysis. First, for more complex tasks—where complexity is measured by lower test accuracy—such as `AQuA-Rat` and `LogiQA`, the decline in explanation quality is more pronounced across all criteria.

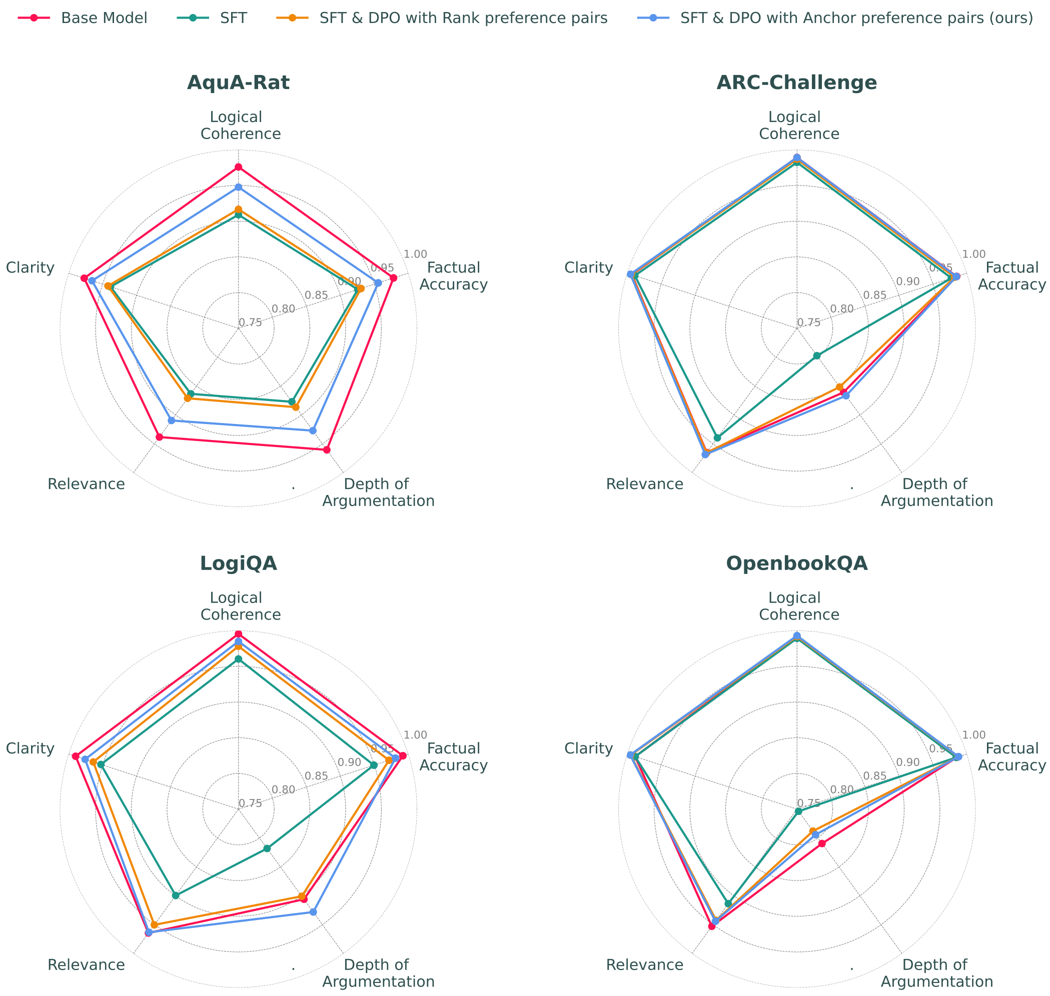

Figure 1: **Average Self-Explanation Scores per Evaluation Criterion**. Average scores for each evaluation criterion used to assess self-explanations, as described in Section 2.1. The scores are provided for all evaluated models across the benchmark datasets.

Second, evaluation dimensions for which the base model originally received lower scores tend to experience a more significant drop in performance after SFT.

4.2.4 Impact of Preference Pairs Category Distribution

We define λ as the proportion of the self-instruct dataset used to align $\mathcal{M}_{\text{Anchor}}$, which corresponds to preference pairs selected under the *consistently-incorrect* or *variable* strategies (see Section 3.3). This metric, λ, provides insight into how the composition of the self-instruct dataset for $\mathcal{M}_{\text{Anchor}}$ differs from that of $\mathcal{M}_{\text{Rank}}$, which selects pairs based solely on scores assigned by judges, following the *consistently-correct* strategy. We evaluated improvements by analyzing the differences in accuracy and Δ_{W-L} between $\mathcal{M}_{\text{Anchor}}$ and $\mathcal{M}_{\text{Rank}}$ in relation to λ (see Figure 2). In both cases, we observed a trend showing that $\mathcal{M}_{\text{Anchor}}$ demonstrates a greater relative improvement compared to $\mathcal{M}_{\text{Rank}}$ as λ increases. This supports our design principle that tailoring strategies based on model behavior is crucial for improving the quality of self-instruct datasets and avoiding the reinforcement of problematic behavior.

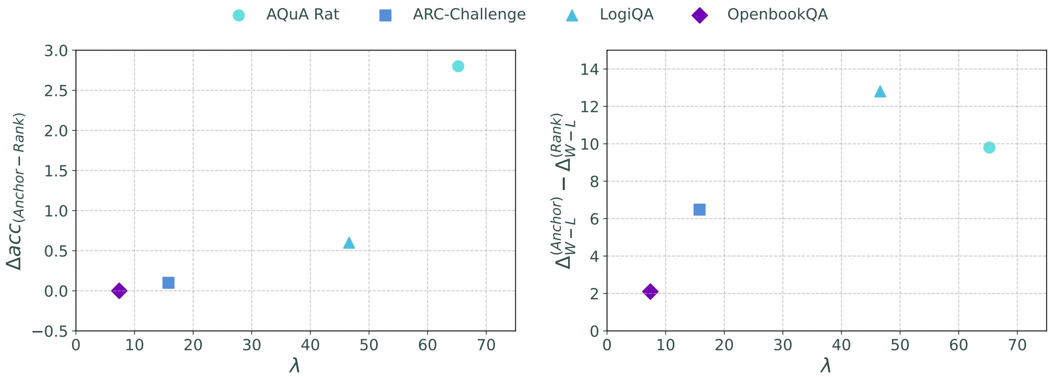

Figure 2: **Impact of Preference Pairs Category Distribution:** Presents the relative improvements in accuracy (*left*) and Δ_{W-L} (*right*) between $\mathcal{M}_{\text{Anchor}}$ and $\mathcal{M}_{\text{Rank}}$ with respect to λ.

5 Related Work

LLM-as-Evaluator: This concept refers to the ability of large language models (LLMs) to evaluate the outputs of other LLMs, a technique commonly referred to as LLM-as-a-Judge. This approach has gained considerable traction in recent years (Dubois et al., 2023; Li et al., 2024; Fernandes et al., 2023; Bai et al., 2023) and is frequently used to assess LLM performance across various downstream tasks (Zheng et al., 2023). It has proven particularly effective in automating evaluations, as demonstrated on platforms like LMSys Chatbot Arena. Key implementations include direct scoring based on specific criteria, pairwise comparisons (Liu et al., 2024), reference-based evaluations, and ensemble methods (Verga et al., 2024). While LLM-as-a-Judge offers scalability and consistency, it can also inherit biases from the evaluation model, potentially amplifying problematic outputs (Huang et al., 2024). Despite these challenges, it remains a valuable tool due to its efficiency and cost-effectiveness in evaluating LLM systems. In our work, we introduce a framework for the qualitative assessment of *self-explanations* using the LLM-as-a-Judge technique, designed to evaluate how effectively a model conveys its reasoning.

Self-Alignment: Several approaches have been developed to improve LLMs without requiring human-annotated feedback. One method involves fine-tuning models using high-quality, self-generated input-output pairs (Wang et al., 2023; Chen et al., 2023; Gulcehre et al., 2023), though this can perpetuate biases in example selection without a clear mechanism for improving selection quality. Another influential approach is Constitutional AI (Bai et al., 2022), where an LLM provides feedback and refines responses, which are then used to train a separate, static reward model. Building on this concept, Yuan et al. (2024) and Wu et al. (2024) proposed using the LLM itself as a dynamic reward model, eliminating the need for a static one. This allows for continuous improvement in both generation and evaluation capabilities through iterative training processes. In our work, we introduce a novel method for creating a self-instruct dataset. Our approach, called *Alignment with Anchor Preference Pairs*, enhances preference pair selection by categorizing model behavior in response to each input prompt and applying tailored strategies for each category. To evaluate a model's consistency for a given input prompt, an anchor—i.e., a ground-truth reference—is required, which we derive from a classification task used as a probing mechanism.

LLM-as-a-Debater: This adversarial approach aims to improve model performance through argumentation. In Perez et al. (2019), debaters are limited to extracting relevant statements from a source text, rather than generating original arguments. Du et al. (2023) extended this concept by involving multiple LLM instances to debate their individual responses over several rounds, eventually converging on a shared final answer. Khan et al. (2024) further developed this approach by using debate-like scenarios to challenge and refine model outputs through simulated arguments. In our work, we adopt the LLM-as-a-Debater approach in the role of a consultant, specifically following Khan et al. (2024), for cases where the model's response to certain input prompts is *consistently incorrect*. This strategy enables the creation of self-instruct examples that avoid reinforcing problematic behavior.

6 Limitations

We acknowledge some limitations in our approach. First, evaluating the model's consistency on a given input prompt requires a anchor—ground truth reference. Consequently, the selection of preference pairs via the anchor strategy relies on a classification task as the probing mechanism, which restricts its applicability. Second, when ranking the quality of self-explanations, we assign equal weights across all evaluation dimensions. This uniform weighting may not accurately reflect the varying significance of different aspects of explanation quality, which can differ depending on the user or specific application. Moreover, this approach may overlook instances where individual explanations degrade in separate criteria, potentially leading to preference pairs where score differences arise from unrelated factors.

Finally, using the base model as the judge during the creation of the self-instruct dataset eliminates the need for a more capable model but introduces a static evaluation process. As the model improves, the judge may fail to capture important evaluation nuances. Iteratively enhancing the judge's capabilities, similar to the approaches in Yuan et al. (2024) and Wu et al. (2024), could help mitigate this issue.

7 Conclusion

In this work, we introduced a methodology for alignment that enhances LLMs' ability to generate high-quality self-explanations, even in the absence of annotated rationale explanations. Our approach provides an end-to-end solution for aligning LLMs on classification tasks, ensuring that they not only produce accurate predictions but also articulate coherent explanations for their decisions. This is achieved through three core components: evaluating the quality of generated explanations, creating self-instruct datasets, and aligning the model. Central to our approach is Alignment with Anchor Preference Pairs, a novel method that refines preference pair selection by categorizing model outputs into three groups: consistently correct, consistently incorrect, and variable. For each category, we apply tailored strategies to construct preference pairs, which are then used in DPO. Our empirical results demonstrate that this method consistently improves explanation quality, reducing the degradation caused by task specialization. Furthermore, we show that using anchor preference pairs outperforms self-alignment methods that rely solely on judge-based evaluations for preference pair selection.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI Feedback, December 2022. URL http://arxiv.org/abs/2212.08073. arXiv:2212.08073 [cs].

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. Benchmarking Foundation Models with Language-Model-as-an-Examiner. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:78142–78167, December 2023.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA Learns Less and Forgets Less. *Transactions on Machine Learning Research*, May 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aloEru2qCG.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering Latent Knowledge in Language Models Without Supervision. *International Conference on Learning Representations (ICLR)*, September 2022.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. AlpaGasus: Training a Better Alpaca with Fewer Data. *International Conference on Learning Representations (ICLR)*, October 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge, March 2018. URL http://arxiv.org/abs/1803.05457. arXiv:1803.05457 [cs].

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving Factuality and Reasoning in Language Models through Multiagent Debate. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S. Liang, and Tatsunori B. Hashimoto. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:30039–30069, December 2023.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André Martins, Graham Neubig, Ankush Garg, Jonathan Clark, Markus Freitag, and Orhan Firat. The Devil Is in the Errors: Leveraging Large Language Models for Fine-grained Machine Translation Evaluation. *Proceedings of the Eighth Conference on Machine Translation*, pp. 1066–1083, December 2023. doi: 10.18653/v1/2023.wmt-1.100. URL https://aclanthology.org/2023.wmt-1.100.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced Self-Training (ReST) for Language Modeling, August 2023. URL http://arxiv.org/abs/2308.08998. arXiv:2308.08998 [cs].

Hui Huang, Yingqi Qu, Hongli Zhou, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. On the Limitations of Fine-tuned Judge Models for LLM Evaluation, June 2024. URL http://arxiv.org/abs/2403.02839. arXiv:2403.02839 [cs].

Nitish Joshi, Javier Rando, Abulhair Saparov, Najoung Kim, and He He. Personas as a Way to Model Truthfulness in Language Models, February 2024. URL http://arxiv.org/abs/2310.18168. arXiv:2310.18168 [cs].

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. Debating with More Persuasive LLMs Leads to More Truthful Answers. *International Conference on Machine Learning (ICML)*, pp. 23662–23733, 2024. ISSN: 2640-3498.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the Effects of RLHF on LLM Generalisation and Diversity, February 2024. URL http://arxiv.org/abs/2310.06452. arXiv:2310.06452 [cs].

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring Faithfulness in Chain-of-Thought Reasoning, July 2023. URL http://arxiv.org/abs/2307.13702. arXiv:2307.13702 [cs].

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. Self-Alignment with Instruction Backtranslation, March 2024. URL http://arxiv.org/abs/2308.06259. arXiv:2308.06259 [cs].

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, July 2017. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 3622–3628, 2020. doi: 10.24963/ijcai.2020/501.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. Aligning with Human Judgement: The Role of Pairwise Preference in Large Language Model Evaluators, March 2024. URL https://arxiv.org/abs/2403.16950v3.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful Chain-of-Thought Reasoning. *International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational*, pp. 305–329, November 2023. doi: 10.18653/v1/2023.ijcnlp-main.20. URL https://aclanthology.org/2023.ijcnlp-main.20.

Andreas Madsen, Sarath Chandar, and Siva Reddy. Are self-explanations from Large Language Models faithful? *Association for Computational Linguistics (ACL)*, pp. 295–337, August 2024a. doi: 10.18653/v1/2024.findings-acl.19. URL https://aclanthology.org/2024.findings-acl.19.

Andreas Madsen, Siva Reddy, and Sarath Chandar. Faithfulness Measurable Masked Language Models. *International Conference on Machine Learning (ICML)*, pp. 34161–34202, July 2024b. URL https://proceedings.mlr.press/v235/madsen24a.html. ISSN: 2640-3498.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2381–2391, October 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260.

Letitia Parcalabescu and Anette Frank. On Measuring Faithfulness or Self-consistency of Natural Language Explanations. *Association for Computational Linguistics (ACL)*, pp. 6048–6089, August 2024. doi: 10.18653/v1/2024.acl-long.329. URL https://aclanthology.org/2024.acl-long.329.

Ethan Perez, Siddharth Karamcheti, Rob Fergus, Jason Weston, Douwe Kiela, and Kyunghyun Cho. Finding Generalizable Evidence by Learning to Convince Q&A Models. *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2402–2411, 2019. doi: 10.18653/v1/D19-1244. URL https://aclanthology.org/D19-1244.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:53728–53741, 2023.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. Branch-Solve-Merge Improves Large Language Model Evaluation and Generation. *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 8352–8370, 2024. doi: 10.18653/v1/2024.naacl-long.462. URL https://aclanthology.org/2024.naacl-long.462.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards Understanding Sycophancy in Language Models, October 2023. URL http://arxiv.org/abs/2310.13548. arXiv:2310.13548 [cs, stat].

Megh Thakkar, Quentin Fournier, Matthew Riemer, Pin-Yu Chen, Amal Zouaq, Payel Das, and Sarath Chandar. A Deep Dive into the Trade-Offs of Parameter-Efficient Preference Alignment Techniques. *Association for Computational Linguistics (ACL)*, pp. 5732–5745, 2024. doi: 10.18653/v1/2024.acl-long.311. URL https://aclanthology.org/2024.acl-long.311.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:74952–74965, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models, May 2024. URL http://arxiv.org/abs/2404.18796. arXiv:2404.18796 [cs].

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions. *Association for Computational Linguistics (ACL)*, pp. 13484–13508, July 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:24824–24837, 2022.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. Meta-Rewarding Language Models: Self-Improving Alignment with LLM-as-a-Meta-Judge. *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2407.19594 [cs].

Haoran Yang, Yumeng Zhang, Jiaqi Xu, Hongyuan Lu, Pheng Ann Heng, and Wai Lam. Unveiling the Generalization Power of Fine-Tuned Large Language Models, March 2024. URL http://arxiv.org/abs/2403.09162. arXiv:2403.09162 [cs].

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-Rewarding Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*, February 2024. arXiv:2401.10020 [cs].

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. Self-Alignment for Factuality: Mitigating Hallucinations in LLMs via Self-Evaluation, June 2024. URL http://arxiv.org/abs/2402.09267. arXiv:2402.09267 [cs].

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A Pairwise Model Evaluation

To compare the performance of two models, denoted as $\mathcal{M}_1$ and $\mathcal{M}_2$, we perform a pairwise evaluation of the self-explanations generated for a given prompt x_i. Each model produces N explanations, and we compare each explanation from $\mathcal{M}_1$ with every explanation from $\mathcal{M}_2$, resulting in N^2 pairwise comparisons.

For a given comparison between the n-th explanation from model $\mathcal{M}_1$ and the m-th explanation from model $\mathcal{M}_2$, where $n, m \in \{1, \ldots, N\}$, we compare the corresponding scores, $s_i^n(\mathcal{M}_1)$ and $s_i^m(\mathcal{M}_2)$. A *win* for $\mathcal{M}_1$ is recorded if the score from $\mathcal{M}_1$ is strictly greater than that from $\mathcal{M}_2$:

$$s_i^n(\mathcal{M}_1) > s_i^m(\mathcal{M}_2)$$

Conversely, a *loss* for $\mathcal{M}_1$ occurs if the score from $\mathcal{M}_1$ is strictly less than the score from $\mathcal{M}_2$:

$$s_i^n(\mathcal{M}_1) < s_i^m(\mathcal{M}_2)$$

A *tie* is defined when both scores are equal:

$$s_i^n(\mathcal{M}_1) = s_i^m(\mathcal{M}_2)$$

For each prompt x_i, we count the total number of wins, losses, and ties across all N^2 comparisons between the explanations from both models. To summarize the performance of the models across the entire dataset, we compute the win rate, tie rate, and loss rate.

The win rate $W(\mathcal{M}_1, \mathcal{M}_2)$ is the average proportion of pairwise comparisons in which model $\mathcal{M}_1$ outperforms model $\mathcal{M}_2$ across all prompts in the set $\mathcal{X}$. It is computed as:

$$W(\mathcal{M}_1, \mathcal{M}_2) = \frac{1}{|\mathcal{X}|} \sum_{x_i \in \mathcal{X}} \left(\frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \nVdash[s_i^n(\mathcal{M}_1) > s_i^m(\mathcal{M}_2)] \right)$$

Here, $\mathcal{X}$ is the set of all prompts, and $\nVdash[\cdot]$ is the indicator function, which returns 1 if the condition inside the brackets is true (i.e., if $\mathcal{M}_1$ wins) and 0 otherwise.

Similarly, we define the tie rate $T(\mathcal{M}_1, \mathcal{M}_2)$ as the proportion of pairwise comparisons where the models perform equally:

$$T(\mathcal{M}_1, \mathcal{M}_2) = \frac{1}{|\mathcal{X}|} \sum_{x_i \in \mathcal{X}} \left(\frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \nVdash[s_i^n(\mathcal{M}_1) = s_i^m(\mathcal{M}_2)] \right)$$

The loss rate $L(\mathcal{M}_1, \mathcal{M}_2)$ captures the proportion of comparisons where $\mathcal{M}_1$ performs worse than $\mathcal{M}_2$:

$$L(\mathcal{M}_1, \mathcal{M}_2) = \frac{1}{|\mathcal{X}|} \sum_{x_i \in \mathcal{X}} \left(\frac{1}{N^2} \sum_{n=1}^{N} \sum_{m=1}^{N} \nVdash[s_i^n(\mathcal{M}_1) < s_i^m(\mathcal{M}_2)] \right)$$

By evaluating the win, tie, and loss rates, we obtain a comprehensive picture of how the two models compare in terms of generating higher-quality self-explanations. This approach accounts for variability in the quality of explanations generated by the models, providing a more nuanced and detailed evaluation than methods relying solely on single-point estimates.

B Inference Parameters

Table 2 summarizes the inference parameters, including temperature and top-k, used for each component, such as the judge, consultant, and sampler.

Table 2: **Inference parameters per component**

Component	**Temperature**	**Top-k**
Judge	0.0	
Consultant	0.5	0.9
Sampler	0.6	0.9

C Judge Component

The judge model $\mathcal{M}_{\text{Judge}}$ evaluates the quality of self-explanation, denoted as ε_i, associated with an input prompt x_i. based on predefined criteria, which are elaborated in Section 2.1. The evaluation process proceed as follows:

1. For each criterion κ, $\mathcal{M}_{\text{Judge}}$ assigns a qualitative verdict $v_{i,\kappa}$ from the set {excellent, satisfactory, needs improvement, unsatisfactory}. The prompt used by $\mathcal{M}_{\text{Judge}}$ is provided in Appendix C.2.

$$\mathcal{M}_{\text{Judge}}(x_i, \varepsilon_i) \to \{v_{i,\kappa}\} \text{ for } \kappa \in \{1, \ldots, K\}$$

2. Each verdict $v_{i,\kappa}$ is mapped to a numerical score $s_{i,\kappa}$ (see Appendix C.1).

3. The overall score for an explanation, s_i, is computed as the sum of scores across all criteria:

$$s_i = \sum_{k=1}^{K} s_{i,k}$$

C.1 Judge Score Mapping

Each verdict $v_{i,\kappa}$, assigned by $\mathcal{M}_{\text{Judge}}$ for criterion κ on self-explanation ε_i corresponding to prompt x_i, is mapped to a numerical score $s_{i,\kappa}$ as follows:

$$s_{i,\kappa} = \begin{cases} 1.0 & \text{if } v_{i,\kappa} = \texttt{\textbf{Excellent}}, \\ 0.8 & \text{if } v_{i,\kappa} = \texttt{\textbf{Good}}, \\ 0.6 & \text{if } v_{i,\kappa} = \texttt{\textbf{Satisfactory}}, \\ 0.2 & \text{if } v_{i,\kappa} = \texttt{\textbf{Needs Improvement}}, \\ 0.0 & \text{if } v_{i,\kappa} = \texttt{\textbf{Unsatisfactory}}. \end{cases}$$

In this mapping, scores $s_{i,\kappa} \in \{1.0, 0.8, 0.6, 0.2, 0.0\}$ represent the numerical evaluation of the verdict, with higher values indicating better performance.

C.2 Judge prompt

Table 3 presents the complete prompt that guides the judge model in evaluating the self-explanations.

Table 3: **Judge prompt**.

Prompt Judge

System: You are an expert judge tasked with evaluating the quality and correctness of a STATEMENT in response to a given QUESTION.

Your goal is to assess whether the STATEMENT accurately and comprehensively answers the QUESTION while providing sound reasoning and clear explanations. Be vigilant for any errors, misleading information, or gaps in the reasoning.

Evaluate the STATEMENT based on the following criteria:

- **Factual Accuracy:** Are all specific claims, data points, and facts in the STATEMENT verifiably correct?
- **Logical Coherence:** Does the STATEMENT follow a clear, logical sequence of reasoning without contradictions, and do all parts logically connect to form a cohesive and unified explanation?
- **Clarity and Comprehensibility:** Is the STATEMENT expressed clearly and precisely, using appropriate terminology and effectively explaining complex ideas?
- **Relevance and Completeness:** Does the STATEMENT fully address all relevant aspects of the QUESTION, ensuring that it directly responds to the specific context or requirements, leaving no key details omitted?
- **Depth of Argumentation:** Does the STATEMENT provide strong reasoning and credible evidence to support its conclusions?

For each criterion, select one of the following assessments:

- **EXCELLENT:** The STATEMENT exemplifies this criterion, setting a high standard.
- **GOOD:** The STATEMENT meets this criterion well, with only minor room for improvement.
- **SATISFACTORY:** The STATEMENT adequately meets this criterion but has noticeable areas for enhancement.
- **NEEDS IMPROVEMENT:** The STATEMENT partially meets this criterion but requires significant revisions.
- **UNSATISFACTORY:** The STATEMENT fails to meet this criterion.

Provide the assessments (EXCELLENT/GOOD/SATISFACTORY/NEEDS IMPROVEMENT/UNSATISFACTORY) for each criterion without additional justification. Respond with the following structure:

- **Factual Accuracy:** [Assessment]
- **Logical Coherence:** [Assessment]
- **Clarity:** [Assessment]
- **Relevance:** [Assessment]
- **Depth of Argumentation:** [Assessment]

User: Here is the question: **[QUESTION]**

Here is the statement you need to evaluate: **[STATEMENT]**

C.3 Consultant Component

In cases where the model $\mathcal{M}_{\text{SFT}}$ behaves consistently incorrectly for the input prompt x_i, we employ the model $\mathcal{M}_{\text{Base}}$ in a consultant role. Specifically, we provide the correct answer y_i to the LLM and request an explanation ε_i supporting this answer.

$$\mathcal{M}_{\text{Debater}}(x_i, y_i) \rightarrow \varepsilon_i$$

C.4 CONSULTANT PROMPT

Table 4: **Consultant prompt**.

Consultant prompt

System: You will act as an expert consultant. You will be shown a QUESTION and one OPTION to argue for. Your goal is to construct convincing arguments supporting your option as correct.

Maintain a scholarly yet assertive tone, focusing on evidence-based analysis while crafting your arguments and answering questions. Avoid phrases like "In summary", "In conclusion", or similar at the end of your argument. Be as concise as possible.

Your argument must fulfill the following criteria:

- **Truthfulness:** Any factual information mentioned in your argument must be true and accurate.
- **Logical Consistency:** Your argument must be logically sound, free of logical fallacies or contradictions.
- **Clarity:** Your argument directly addresses and answers the question. It is clear, concrete, concise, and well-structured.

User: Here is the question: **[QUESTION]**

Here is option you need to back up: **[OPTION]**

D PROMPTS INFERENCE PER TASK

Table 5: **Prompt for LogiQA task**.

Prompt for LogiQA task

System: You will be presented with a CONTEXT passage and a corresponding QUESTION with four answer CHOICES. Carefully read the passage to understand its content. Then, read the QUESTION and CHOICES thoroughly. Choose the correct CHOICE and explain your reasoning.

Your response will consist of two parts: an EXPLANATION followed by your selected CHOICE.

Enclose your explanation within tags as follows:
<explanation>[Your EXPLANATION here]</explanation>

Enclose your chosen choice (e.g., if the question has only 4 choices, then A, B, C, or D) within tags as follows:
<choice>[Your CHOICE here]</choice>

User: Context: **[CONTEXT]**

Question: **[QUESTION]**

Choices: **[CHOICES]**

D.1 Prompt for AQuA-Rat task

Table 6: **Prompt for AQuA-Rat task**.

Prompt for AQuA-Rat task

System: You will be given a QUESTION along with multiple answer CHOICES, involving a math problem that requires step-by-step reasoning to determine the correct answer. Carefully read the QUESTION and CHOICES. Choose the correct CHOICE and explain your reasoning.

Your response will consist of two parts: an EXPLANATION followed by your selected CHOICE.

Enclose your explanation within tags as follows:
<explanation>[Your EXPLANATION here]</explanation>

Enclose your chosen choice (e.g., if the question has only 4 choices, then A, B, C, or D) within tags as follows:
<choice>[Your CHOICE here]</choice>

User: Context: **[CONTEXT]**

Question: **[QUESTION]**

Choices: **[CHOICES]**

D.2 Prompt for ARC-Challenge task

Table 7: **Prompt for ARC-Challenge task**.

Prompt for ARC-Challenge task

System: You will be presented a QUESTION with multiple answer CHOICES. Carefully read the QUESTION and CHOICES. Choose the correct CHOICE and explain your reasoning.

Your response will consist of two parts: an EXPLANATION followed by your selected CHOICE.

Enclose your explanation within tags as follows:
<explanation>[Your EXPLANATION here]</explanation>

Enclose your chosen choice (e.g., if the question has only 4 choices, then A, B, C, or D) within tags as follows:
<choice>[Your CHOICE here]</choice>

User: Context: **[CONTEXT]**

Question: **[QUESTION]**

Choices: **[CHOICES]**

D.3 Prompt for OpenbookQA task

Table 8: **Prompt for OpenbookQA task**.

Prompt for OpenbookQA task

```
System: You will be presented a QUESTION with multiple answer CHOICES. Carefully read
the QUESTION and CHOICES. Choose the correct CHOICE and explain your reasoning.

Your response will consist of two parts: an EXPLANATION followed by your selected
CHOICE.

Enclose your explanation within tags as follows:
<explanation>[Your EXPLANATION here]</explanation>

Enclose your chosen choice (e.g., if the question has only 4 choices, then A, B, C,
or D) within tags as follows:
<choice>[Your CHOICE here]</choice>
```

```
User: Context: [CONTEXT]

Question: [QUESTION]

Choices: [CHOICES]
```

E Generated Instructions

Table 9: **Distribution of anchor categories:** This table presents the distribution of the categories—Consistently Correct (CC), Consistently Incorrect (CI), and Variable (V)—across datasets used during the DPO alignment phase of $\mathcal{M}_{\text{Anchor}}$.

Dataset	Category	Samples	Ratio (%)
AQuA-Rat	V	1196	41.17
	CC	1010	34.77
	CI	699	24.06
ARC-Challenge	V	62	8.09
	CC	645	84.20
	CI	59	7.70
LogiQA	V	1251	26.86
	CC	2487	53.39
	CI	920	19.75
OpenbookQA	V	176	5.13
	CC	3178	92.60
	CI	78	2.27