Reasons to Reject? Aligning Language Models with Judgments

Anonymous ACL submission

Abstract

As humans, we consistently interact with our peers and receive feedback in the form of natural language. This language feedback allows us to maintain appropriate behavior, and rectify potential errors. The question arises naturally: can we use language feedback to align large language models (LLMs)? In contrast to previous research that aligns LLMs with scalar rewards, we present the first systematic exploration of alignment through the lens of language feedback (i.e., judgment). We start with an in-depth investigation of potential methods that can be adapted for aligning LLMs with judgments, revealing that these methods cannot fully capitalize on judgments. To facilitate more effective utilization of judgments, we propose a novel framework, Contrastive Unlikelihood Training (CUT), which allows for fine-grained inappropriate content detection and correction based on judgments. Our results show that, with merely 1317 off-the-shelf judgment data, CUT (LLaMA2-13b) can beat the 175B DaVinci003 and surpass the best baseline by 52.34 points on AlpacaEval. CUT (LLaMA2-chat-13b) can also align LLMs in an iterative fashion using up-to-date model-specific judgments, improving performance from 81.09 to 91.36 points on AlpacaEval. Further analysis suggests that judgments hold greater potential than rewards in LLM alignment.

1 Introduction

Large language models (LLMs) acquire extensive knowledge and remarkable reasoning capabilities through large-scale pre-training (Brown et al., 2020; Du et al., 2022; Touvron et al., 2023). To unleash the power of pre-trained LLMs for addressing real-world applications, it is crucial to ensure LLMs can follow human values (Ouyang et al., 2022). This process, known as alignment, has the potential to pave the way for a future in which artificial intelligence (AI) serves as a helpful and reliable ally for humanity (Wang et al., 2023b).

Figure 1 shows three typical paradigms to achieve alignment. The most straightforward one is learning from demonstrations, wherein demonstrations of desired responses to a set of instructions are collected to fine-tune LLMs (Wei et al., 2022; Ouyang et al., 2022). However, the performance gains diminish rapidly when scaling up the data size (Zhou et al., 2023). On the other hand, learning from feedback offers a more scalable approach (Ouyang et al., 2022; Bai et al., 2022a). One key advantage of feedback over demonstrations is that feedback can convey both positive and negative aspects, enabling the model to discern desirable and undesirable outcomes. In addition, feedback is tailored to the current model, adhering to the principle of teaching according to the learner's aptitude.

Prior research on learning from feedback primarily focuses on value feedback (i.e., scalar rewards). Reinforcement learning (RL) techniques, particularly PPO algorithm (Schulman et al., 2017), are employed to optimize an LLM to maximize the rewards of its generated responses. However, PPO is known to be complex and often unstable (Zheng et al., 2023), which has prompted numerous efforts to simplify or stabilize the training process (Ramamurthy et al., 2023; Peng et al., 2023b; Dong et al., 2023). Another strand of work, named Hindsight (Zhang et al., 2023; Liu et al., 2023a), transforms scalar rewards to language instructions and employs supervised training on the updated data.

Language feedback (i.e., judgment) is another kind of feedback that offers nuanced commendations and critiques through natural language expressions. Unlike scalar rewards, which are information-sparse for solely indicating the goodness of a response, judgments can elucidate the specific aspects that are good or bad, the rationale behind their evaluation, and suggestions for improvement. The above advantages suggest that aligning LLMs with judgments can be more beneficial (Saunders et al., 2022). However, current

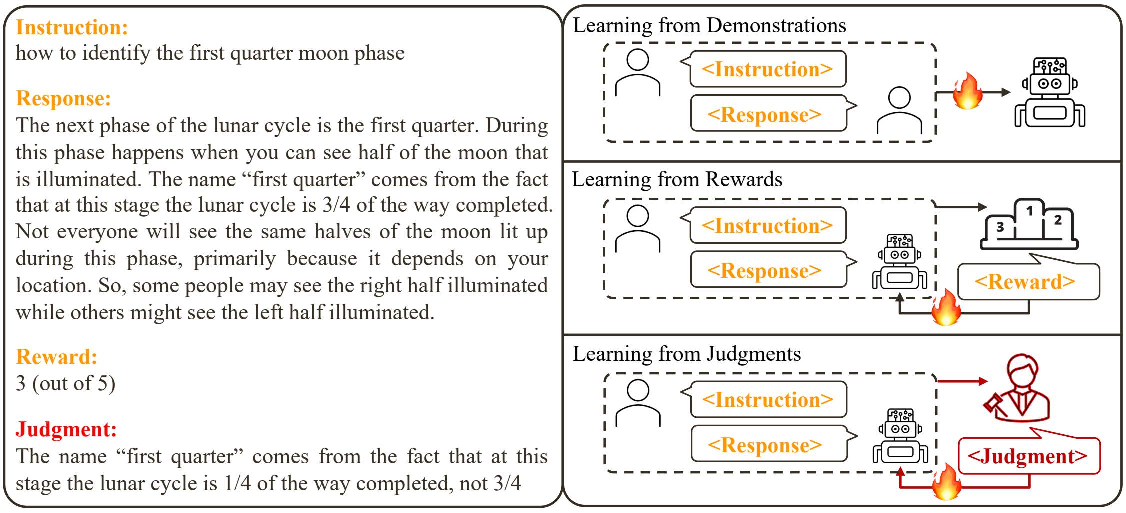


Figure 1: The illustration of three paradigms for aligning LLMs.

approaches merely use judgments to prompt LLMs for an improved response, which is subsequently employed as a new demonstration for supervised training (Bai et al., 2022b; Scheurer et al., 2022, 2023). This indirect utilization of judgments suffers from the incapability to learn from mistakes, which is the core spirit of learning from feedback.

In this study, we present an extensive investigation of potential methods that can be adapted for *aligning LLMs with judgments*. To facilitate a comprehensive aligning process, we propose a novel framework, Contrastive Unlikelihood Training (CUT), that enables fine-grained inappropriate content detection and correction based on judgments. CUT detects inappropriate content in a response by contrasting its generation probabilities under aligned and misaligned conditions and further penalizes the inappropriate content with unlikelihood training (UT) (Welleck et al., 2020).

We carry out experiments for both offline and online alignment, wherein the target LLM learns from the off-the-shelf judgments and the judgments derived from self-generated responses, respectively. Extensive results on offline alignment demonstrate the effectiveness of CUT in learning from judgments in both cold-start (using unaligned base LLMs such as LLaMA2) and warm-start (using aligned base LLMs such as LLaMA2-chat) scenarios. Notably, when trained with only 1317 offline judgment data, CUT (LLaMA2-13b) attains a winning rate of 62.56 and outperforms the best baseline by 52.34 points on AlpacaEval. Furthermore, our online alignment experiments show that CUT is capable of iteratively refining LLMs (LLaMA2-chat-13b) using model-specific judgments, with a steady performance improvement from 81.09 to 91.36 points on AlpacaEval. Our analysis comparing rewards and judgments suggests that aligning LLMs with judgments offers significant potential and warrants future research.

Our contributions can be summarized as follows: 1) We present the first systematic exploration of aligning LLMs with judgments. 2) We introduce a novel framework, CUT, that facilitates the alignment of LLMs through fine-grained inappropriate content detection and correction based on judgments. 3) Our results showcase the effectiveness of CUT in aligning LLMs across cold-start and warm-start scenarios, generalist and specialist applications, as well as offline and online settings. 4) Our analysis indicates that judgments hold greater potential over rewards for aligning LLMs.

2 Related Work

Existing approaches for learning from feedback can be classified into two distinct categories: prompting and fine-tuning, differentiated by whether updates to the LLMs' parameters are absent or present.

Prompting. Prompting does not alter the parameters of LLMs. Instead, it leverages judgments on previous responses to elicit improved responses from LLMs (Welleck et al., 2022; Akyurek et al., 2023). Judgments can be sourced from diverse aspects (Nathani et al., 2023; Yu et al., 2023) and the refinement process can be iterated multiple times (Yang et al., 2022; Peng et al., 2023a; Madaan et al., 2023). However, these methods consume more computation than single-pass generation and usually rely on the in-context learning capabilities of the LLMs (Brown et al., 2020; Liu et al., 2023b).

Fine-tuning. Fine-tuning aims to directly train a better LLM. Scalar rewards have been extensively used through RL, particularly PPO (Schulman et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Yang et al., 2023). However, these RL approaches are notoriously unstable and complex (Zheng et al., 2023). To stabilize RL, Ramamurthy et al. (2023) propose to reduce the action space through truncation and Peng et al. (2023b) employ an advantage model and selective rehearsal. In addition, many studies aim to design simpler alternatives to RL. Dong et al. (2023); Touvron et al. (2023) treat rewards as a ranking criterion and simply train models using the best model-generated responses. There are also attempts to leverage the results of prompting for training a better model. That is, the improved response elicited by the judgment is employed as new training data (Bai et al., 2022b; Scheurer et al., 2022, 2023). However, these methods still suffer from the incapability to learn from mistakes. Rafailov et al. (2023); Yuan et al. (2023); Song et al. (2023) demonstrate that LLMs themselves can be used as reward functions and derive different training objectives to eliminate the need for RL. Zhang et al. (2023); Liu et al. (2023a) relabel the input using the reward received by the response, referred to as *Hindsight*. Hindsight allows LLMs to generate responses of different qualities.

3 Preliminaries

In this section, we first lay out a formal problem definition of *aligning LLMs with judgments* and then present a survey of three potential methods that can be adapted for tackling this problem.

3.1 Problem Setting

Suppose that there is a set of instruction-response-judgment triplets $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{j})$, where the instruction $\boldsymbol{x} = [x_1, \dots, x_M]$, the response $\boldsymbol{y} = [y_1, \dots, y_N]$, and the judgment $\boldsymbol{j} = [j_1, \dots, j_Q]$ are token sequences of length M, N, and Q, respectively. The response may exhibit certain flaws or be considered entirely satisfactory. The judgment provides an analysis of the strengths and weaknesses of the response, which can be drafted either by humans or AI models (Akyurek et al., 2023; Li et al., 2023). The goal of aligning LLMs with judgments is to enable LLMs to retain appropriate behaviors mentioned in the strengths, and more importantly, address the weaknesses to prevent future misbehavior.

Depending on whether the responses $\boldsymbol{y}$ are from the LLM to be aligned, the learning process can be classified into two distinct types: *offline alignment* and *online alignment*. In offline alignment, the target LLM learns from an off-the-shelf, model-agnostic dataset. Conversely, in online alignment, the target LLM reflects on its own outputs through direct interactions with a judge. This online alignment process can be conducted iteratively, akin to how humans continuously improve their skills by receiving ongoing feedback from others over time.

3.2 Potential Solutions

Forward Prediction refers to sequentially predicting the response and its judgment, which was originally proposed in dialogue generation (Weston, 2016; Li et al., 2017). It can be seamlessly adapted to our problem. Specifically, the LLM is trained with the maximum likelihood estimation (MLE) objective to first generate the response $\boldsymbol{y}$ based on the instruction $\boldsymbol{x}$ and subsequently generate the judgment $\boldsymbol{j}$ based on the combined sequence $[\boldsymbol{x}, \boldsymbol{y}]$.

$$L_f = -\frac{1}{N}\sum_t \log p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) - \frac{1}{Q}\sum_t \log p(j_t|\boldsymbol{j}_{<t}, \boldsymbol{y}, \boldsymbol{x}) \tag{1}$$

Imitation Learning from Language Feedback (ILF) asks the LLM to refine the initial response $\boldsymbol{y}$ given the feedback $\boldsymbol{j}$ (Bai et al., 2022b; Scheurer et al., 2022, 2023). The improved response $\hat{\boldsymbol{y}}$, paired with the initial instruction $\boldsymbol{x}$, is used to fine-tune the LLM under the MLE objective.

$$\begin{aligned} \hat{\boldsymbol{y}} &= \mathbf{LLM}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{j}) \\ L_i &= -\frac{1}{N}\sum_t \log p(\hat{y}_t|\hat{\boldsymbol{y}}_{<t}, \boldsymbol{x}) \end{aligned} \tag{2}$$

Hindsight rewrites the instruction $\boldsymbol{x}$ based on the scalar rewards received by the response $\boldsymbol{y}$ (Zhang et al., 2023; Liu et al., 2023a). For instance, if a response receives a scalar reward below a certain threshold, the phrase “generate a good answer" is appended to the original instruction. This approach can be naturally extended to our problem setting. Concretely, the LLM is trained to generate the response $\boldsymbol{y}$ conditioned on the sequence $[\boldsymbol{x}, \boldsymbol{j}]$.

$$L_h = -\frac{1}{N}\sum_t \log p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}, \boldsymbol{j}) \tag{3}$$

However, in forward prediction, learning to generate judgments does not necessarily translate into enhanced response generation, given that response generation precedes judgment generation. ILF only uses the positive data (i.e., the improved responses),

	Instruction: $\boldsymbol{x}$	Response: $\boldsymbol{y}$	Judgment: $\boldsymbol{j}$	$\boldsymbol{x} \rightarrow \boldsymbol{y}$	$[\boldsymbol{x}, \boldsymbol{j}] \rightarrow \boldsymbol{y}$
Align-P	James buys 5 packs of beef that are 4 pounds each. The price of beef is $5.50 per pound. How much did he pay?	He bought 5 * 4 = 20 pounds of beef. So he paid 20 * 5.5 = $110.	Your response to the instruction is satisfactory.	✓	✓
Align-N	James buys 5 packs of beef that are 4 pounds each. The price of beef is $5.50 per pound. How much did he pay?	Each pack was 5 pounds and it cost 5.50. So 5 * 5.50 = $27.50.	The answer forgets to multiply the total amount of pounds of beef (5*4).	✗	✓
Misalign	James buys 5 packs of beef that are 4 pounds each. The price of beef is $5.50 per pound. How much did he pay?	Each pack was 5 pounds and it cost 5.50. So 5 * 5.50 = $27.50.	Your response to the instruction is satisfactory.	✗	✗

Table 1: The illustration of three categories of alignment data. $\boldsymbol{x} \rightarrow \boldsymbol{y}$ and $[\boldsymbol{x}, \boldsymbol{j}] \rightarrow \boldsymbol{y}$ indicate if the response aligns with the instruction or the combination of instruction and judgment, respectively.

limiting its capacity to spot and rectify weaknesses underscored in judgments. Hindsight employs unsatisfactory responses as MLE targets, which inevitably increases the risk of generating unsatisfactory responses. In summary, we contend that existing methods cannot fully capitalize on judgments, which motivates us to design a better solution.

4 Contrastive Unlikelihood Training

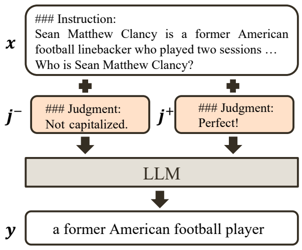
To overcome the limitations mentioned in § 3, we propose CUT, a novel fine-tuning framework to align LLMs with judgments. The core idea of CUT is summarized as **Learning from Contrasting**. We contrast the response generation under different conditions to shed light on the appropriate behavior that the LLM should keep, as well as the specific content necessitating adjustments. Based on these insights, we use MLE training for appropriate content and UT for inappropriate content.

4.1 Incorporating Judgments for Alignment

We call an instruction-response pair “aligned" if the response follows the instruction faithfully and satisfies human expectations $\boldsymbol{x} \rightarrow \boldsymbol{y}$. Otherwise, a judgment describes the errors or deficiencies present in the response. Assuming the task is to generate a response that intentionally fulfills the judgment, it can be inferred that the response always aligns with the combined input of instruction and judgment $[\boldsymbol{x}, \boldsymbol{j}] \rightarrow \boldsymbol{y}$. Based on the idea, we construct three types of alignment data, depicted in Table 1.

Align-P: The LLM produces a satisfactory response $\boldsymbol{y}$ to the instruction $\boldsymbol{x}$. Therefore, a positive judgment $\boldsymbol{j}$ is conferred to praise the commendable performance. The response $\boldsymbol{y}$ is aligned with the instruction $\boldsymbol{x}$ as well as the combined input $[\boldsymbol{x}, \boldsymbol{j}]$.

Align-N: The LLM makes some mistakes in its generation, resulting in an unsatisfactory response



y	$p(y\|x,j^-)$	$p(y\|x,j^+)$	Objective
a	0.82	0.03	UT
former	0.94	0.94	MLE
American	0.76	0.93	MLE
football	0.99	0.99	MLE
player	0.02	0.03	MLE
</s>	0.89	0.85	MLE

Figure 2: Generation probability of identical output text under Align-N (left) and Misalign (right) contexts.

$\boldsymbol{y}$. Consequently, a negative judgment $\boldsymbol{j}$ details the corresponding critiques. For Align-N, $\boldsymbol{y}$ is not aligned with original instruction $\boldsymbol{x}$. However, when considering $\boldsymbol{x}$ and $\boldsymbol{j}$ as a whole, $\boldsymbol{y}$ is indeed aligned with the combined input $[\boldsymbol{x}, \boldsymbol{j}]$.

Misalign: The authentic negative judgment in Align-N is substituted with a fake positive judgment $\boldsymbol{j}$. In this case, the response $\boldsymbol{y}$ is not aligned with either the original instruction $\boldsymbol{x}$ or the combination of instruction and judgment $[\boldsymbol{x}, \boldsymbol{j}]$.

4.2 Learning from Contrasting

With the above three categories of alignment data. We can deduce two notable contrasts that provide valuable insights to guide the alignment of LLMs.

Align-N vs. Misalign: The major difference between these two is that they show opposite polarities in the task of $[\boldsymbol{x}, \boldsymbol{j}] \rightarrow \boldsymbol{y}$. Thanks to the strong in-context learning capabilities of LLMs, the alignment flip from Align-N (aligned) to Misalign (misaligned) is often accompanied by decreased generation probabilities of the response, particularly for tokens that exhibit a strong correlation with the authentic negative judgment. Figure 2 presents a simple example, wherein the response commits a minor capitalization issue. The LLM assigns a considerably higher probability for “a" when taking the authentic negative judgment $\boldsymbol{j}^-$ instead of the fake positive judgment $\boldsymbol{j}^+$ as additional input, precisely at the point where the LLM commits the error.

To take advantage of the above contrast, we feed Align-N and Misalign examples to the LLM to get token generation probabilities $p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}, \boldsymbol{j}^-)$ and $p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}, \boldsymbol{j}^+)$ separately. We consider the tokens that display a substantially increased generation probability when conditioned on $\boldsymbol{j}^-$ compared to $\boldsymbol{j}^+$ as inappropriate tokens (e.g., "a" in Figure 2). Concretely, the following criterion is adopted:

$$U = \{t \mid p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}, \boldsymbol{j}^-) - \lambda \cdot p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}, \boldsymbol{j}^+) > 0\} \tag{4}$$

where λ is a hyperparameter to tradeoff the precision and recall of detecting inappropriate tokens.

We apply the UT objective (Welleck et al., 2020) on the identified inappropriate tokens for pushing the LLM to explore alternative generations. For other tokens, we use the standard MLE loss:

$$\begin{aligned} L_1 = -\frac{1}{N}(&\sum_{t \notin U} \log p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) \\ &+ \sum_{t \in U} \alpha \log(1 - p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}))) \end{aligned} \tag{5}$$

where α is to control the scale of unlikelihood loss.

Align-P vs. Align-N: Despite both Align-P and Align-N are aligned in terms of $[\boldsymbol{x}, \boldsymbol{j}] \rightarrow \boldsymbol{y}$, only Align-P is aligned when solely considering the instruction ($\boldsymbol{x} \rightarrow \boldsymbol{y}$). Essentially, it suggests that the LLM should output different responses depending on whether a negative judgment is incorporated or not. Therefore, the comparison provides valuable information for the LLM to discern satisfactory and unsatisfactory responses. Specifically, we train on this comparison with the following MLE objective:

$$\begin{aligned} L_2 = &-\frac{\mathbb{1}(\boldsymbol{x} \rightarrow \boldsymbol{y})}{N} \sum_t \log p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) \\ &- \frac{(1 - \mathbb{1}(\boldsymbol{x} \rightarrow \boldsymbol{y}))}{N} \sum_t \log p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{j}, \boldsymbol{x}) \end{aligned} \tag{6}$$

where $\mathbb{1}(\boldsymbol{x} \rightarrow \boldsymbol{y})$ is an indicator function that returns 1 if x and y are aligned, and 0 otherwise.

Finally, the overall loss of CUT combines the losses from the two contrasts: $L_{\text{CUT}} = L_1 + L_2$.

4.3 Relation to Prior Solutions

We discuss the connections of CUT to prior solutions of learning from judgments.

Forward Prediction hopes that the judgment generation could indirectly boost its response generation abilities. In contrast, CUT directly utilizes judgments to teach the LLM how to generate satisfactory responses and avoid unsatisfactory ones.

ILF assumes judgments can always elicit improved responses and solely learn from such pseudo-aligned instruction-response pairs. Conversely, CUT can directly learn from misaligned data.

Hindsight learns to generate responses of different qualities at the risk of increasing the likelihood of unsatisfactory responses. In comparison to Hindsight, CUT mitigates this issue by incorporating UT objectives for inappropriate tokens.

5 Experiments

We implement CUT in two alignment settings, namely, online alignment and offline alignment, to demonstrate the overall effectiveness of CUT. Subsequently, to highlight the immense potential of aligning LLMs with judgments, we establish a comparison between learning from rewards and learning from judgments. The details of the model implementations are provided in Appendix A.1.

5.1 Offline Alignment

The offline setting utilizes off-the-shelf instruction-response-judgment triplets for alignment. This aims to check the feasibility of the CUT in learning from judgments prior to initiating the costly process of model-specific judgment annotation.

Tasks. We experiment on a general instruction-following task, and a specific NLP task (summarization). For **Instruction following**, we train models with 1317 examples from Shepherd (Wang et al., 2023a) and evaluate models on four ranking-based and one generation-based LLM benchmarks. The ranking benchmarks are 25-shot ARC, 10-shot HellaSwag, 5-shot MMLU, and 0-shot TruthfulQA from the Open LLM Leaderboard (Gao et al., 2021). The generation benchmark is AlpacaEval, where GPT4 judges the winning rate of the responses generated by our models against DaVinci003. For **Summarization**, we use the dataset with judgment annotations from Saunders et al. (2022). We train models on 10827 training examples and report ROUGE scores (Lin, 2004) on 1939 test examples. See Appendix A.3 for more details.

Setup. We experiment with two base models, LLaMA2-13b and LLaMA2-chat-13b, aiming to demonstrate the efficacy of CUT in both cold-start and warm-start scenarios, respectively. The baselines include the base model without further fine-tuning, and the three judgment-based alignment methods: ILF, Forward Prediction, and Hindsight.

	Model	Objective	ARC	HellaSwag	MMLU	TruthfulQA	Avg.	AlpacaEval
LLaMA2	Base	-	59.72	81.39	54.97	36.28	58.09	1.87
	ILF	MLE	58.36	81.15	53.76	37.03	57.58	4.01
	Forward Prediction	MLE	56.91	81.03	54.35	34.28	56.64	7.11
	Hindsight	MLE	58.11	81.33	55.33	35.61	57.60	10.22
	CUT (ours)	MLE+UT	**59.81**	**81.60**	**55.74**	**49.36**	**61.62**	**62.56**
LLaMA2-chat	Base	-	58.02	79.89	54.52	45.44	59.47	81.09
	ILF	MLE	58.36	**81.15**	53.76	45.65	59.73	79.31
	Forward Prediction	MLE	52.22	78.16	53.06	37.69	55.28	33.21
	Hindsight	MLE	53.92	78.58	54.15	39.01	56.42	36.67
	CUT (ours)	MLE+UT	**58.45**	79.32	**54.82**	**48.84**	**60.36**	**87.24**

Table 2: Results on the general instruction-following task. The Objective column denotes the fine-tuning objective.

	Model	rouge1	rouge2	rougeL	rougeLsum
LLaMA2	Base	12.91	6.33	10.10	10.87
	ILF	28.51	16.68	25.36	25.44
	Forward Prediction	42.42	28.02	38.45	38.51
	Hindsight	38.33	25.49	35.26	35.29
	CUT (ours)	**44.98**	**28.33**	**39.67**	**39.72**
LLaMA2-chat	Base	29.21	15.00	22.78	23.44
	ILF	39.21	27.93	34.35	34.66
	Forward Prediction	42.44	28.12	38.48	38.46
	Hindsight	41.02	27.48	37.42	37.46
	CUT (ours)	**45.35**	**28.60**	**39.98**	**40.05**

Table 3: Results on the summarization task.

Model	Generalist	Specialist
LLaMA2-chat	45.44	23.44
CUT	48.84	40.05
- L_1	39.01	37.46
- first part of L_2	-	27.73
- second part of L_2	46.42	33.60
- Inappropriate Token Detection	0	0

Table 4: Effect of CUT designs. We report the results on TruthfulQA (Acc.) and summarization test set (rougeLsum) for general instruction-following (Generalist) and Summarization (Specialist) respectively. "-" indicates no Align-P examples in the Generalist training set.

Results. The results of the general instruction-following and summarization are presented in Table 2 and 3, respectively. For cold-start scenarios (LLaMA2 as the base model), CUT improves the winning rate on AlpacaEval from 1.87 to 62.56 and surpasses the best baseline (Hindsight) by 52.34 points. This is particularly noteworthy as the resulting 13B model, fine-tuned with merely 1317 examples, can beat 175B DaVinci003. Moreover, CUT improves the base model by 13.08 points on TruthfulQA. This implies that CUT can effectively mitigate hallucinations. Conversely, most baselines experience considerable performance drops on TruthfulQA. This is likely due to their application of the MLE objective on error-prone responses, which reduces factuality in response generation. In terms of ARC, HellaSwag, and MMLU, CUT's performance remains competitive with the base model, indicating CUT suffers less from the alignment tax problem (Ouyang et al., 2022). For single NLP task (i.e., summarization) experiments, CUT surpasses the best baseline (i.e., Forward Prediction) by 1.21 rougeLsum scores. Overall, the results show that CUT is effective in transforming LLMs into both performant generalist and specialist models.

The performance improvements of warm-start scenarios (LLaMA2-chat as the base model) are consistent with the cold-start ones, showcasing the efficacy of CUT in learning from judgments in both cold-start and warm-start scenarios. Interestingly, ILF outperforms Forward Prediction and Hindsight on AlpacaEval in warm-start scenarios but performs worse in cold-start scenarios. This may be due to that ILF heavily relies on the base model in producing high-quality improved responses, making it less effective in cold-start scenarios.

Ablation Study. To investigate the effectiveness of two contrasts employed by CUT, we perform ablation studies by eliminating certain training signals. The results are shown in Table 4. Removing the contrast between Align-N and Misalign (- L_1) substantially reduces the performance of TruthfulQA. This finding highlights that the UT objective plays a crucial role in mitigating hallucinations. The exclusion of the contrast between Align-P and Align-N can be implemented in two ways. We can either remove the first part or the second part of L_2. As seen, the impact of removing Align-P is more pronounced than removing Align-N on the summarization task. This may be attributed to the necessity of positive examples for adapting the LLM to a specific task. Furthermore, we introduce an additional

Model	#J	ARC	HellaSwag	MMLU	TruthfulQA	AlpacaEval
LLaMA2-chat	-	58.02	**79.89**	54.52	45.44	81.09
CUT (offline)	1317	**58.45**	79.32	54.82	48.84	87.24
CUT 1+ (online iteration-1)	1000	57.85	79.34	54.75	49.98	89.81
CUT 2+ (online iteration-2)	1000	58.11	79.13	54.92	50.84	90.55
CUT 3+ (online iteration-3)	1000	58.36	79.04	55.04	51.54	90.99
CUT 4+ (online iteration-4)	1000	58.11	78.88	55.03	51.72	**91.36**
CUT 5+ (online iteration-5)	1000	58.02	78.84	**55.19**	**51.92**	90.61

Table 5: The results of online iterative alignment. #J denotes the number of judgment data used in each iteration.

ablated variant in which the inappropriate token detection (Eq. 4) is omitted (- Inappropriate Token Detection). Concretely, we simply apply UT for all tokens in misaligned responses instead. Intriguingly, we find that this approach fails to converge during training. This observation underscores the importance of inappropriate token detection.

5.2 Online Alignment

In this section, we move to a more pragmatic scenario where the target LLM directly learns from the judgments associated with its own responses.

5.2.1 Iterative Alignment

Setup. As mentioned in § 3.1, the online alignment process can be conducted iteratively, akin to how humans continuously refine their behaviors through ongoing feedback. Specifically, we apply the following three steps repeatedly:

- **Step 1:** Collect a set of instructions $\boldsymbol{x}$, and obtain the responses $\boldsymbol{y}$ from the target model.
- **Step 2:** Annotate judgments $\boldsymbol{j}$ for the responses.
- **Step 3:** Apply CUT to fine-tune the target model with $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{j}\}$.

where the target LLM is LLaMA2-chat. In each iteration, we sample 1000 instructions from Stanford Alpaca (Taori et al., 2023). We ask GPT4 to draft judgments, which has been proven to produce high-quality annotations (Cui et al., 2023). Annotation details are elaborated in Appendix A.2. Note that most responses from LLaMA2-chat receive positive judgments, resulting in a large proportion of Align-P examples. We found downsampling Align-P examples is beneficial to the online alignment (Appendix A.4). We evaluate models on ARC, HellaSwag, MMLU, TruthfulQA, and AlpacaEval.

Results. Table 5 shows the results of online iterative alignment. In the first iteration, online alignment exhibits superior performance over offline alignment on both TruthfulQA and AlpacaEval. This observation implies that model-specific judgments are more effective for alignment. More importantly, the alignment continues to improve with more iterations, where the performance rises from 81.09 to 91.36 on AlpacaEval after four iterations. However, the performance improvement ceases at the fifth iteration. We speculate two possible explanations for this occurrence: (1) the judgments provided by GPT-4 contain certain inaccuracies, making them insufficient to effectively align a strong target model like our CUT 4+. (2) The target model may exhibit a knowledge deficiency in specific domains, such as mathematics and science, which cannot be adequately addressed through judgments. We also provide a case study in Appendix A.5.

5.2.2 Training A Judgment Model

In the previous experiments, we show that CUT is effective in aligning LLMs with judgments annotated by humans or GPT4. However, both annotations can be expensive or infeasible. Therefore, we investigate the possibilities of developing an AI judge based on the current open-source LLMs.

Setup. we train AI judges with different amounts of judgment data $\{3000, 5000\}$ collected in § 5.2.1. Then, we sample 1000 new instructions from Stanford Alpaca, obtain the corresponding responses from the target model (i.e., LLaMA2-chat), and label judgments with our AI judges. These new judgment triplets are used to align the target model.

Results. Figure 3 shows that AI judge-5000, trained with 5000 judgment data, is beneficial for aligning the target LLM, which leads to improvements of 1.6 and 3.41 points on TruthfulQA and AlpacaEval respectively. In contrast, AI Judge-3000, using a smaller training dataset, shows limited effectiveness. The comparison suggests that training a capable AI judge necessitates a moderate number of high-quality training instances. As a result, it is feasible to train AI judges to align the LLM. However, the quality of the AI judge remains a crucial factor in determining the success of this endeavor.

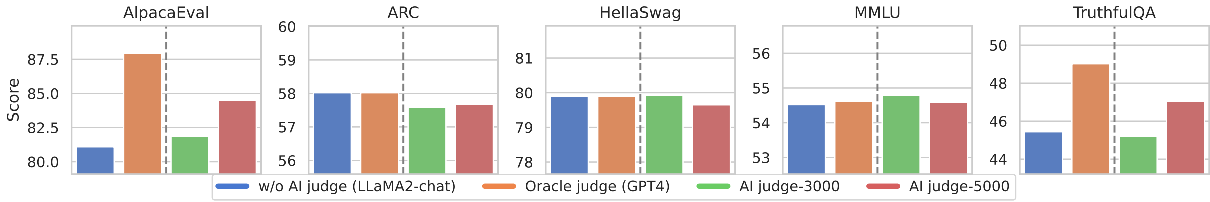


Figure 3: The results of online alignment with different AI judges.

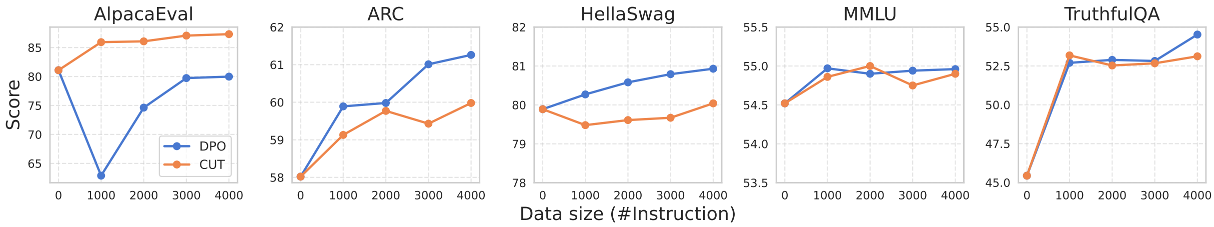


Figure 4: Comparison between reward-based DPO and judgment-based CUT in aligning LLMs.

5.3 Judgment vs. Reward

Our work primarily focuses on aligning LLMs with judgments, whereas most prior research explores rewards. In this section, we aim to provide a direct comparison between these two paradigms. However, note that it is hard to conduct a fair comparison due to the distinct data formats and the potential variation in data quality.

Setup. We compare judgment-based CUT with the state-of-the-art reward-based DPO (Rafailov et al., 2023). To maximize fairness, we leverage UltraFeedback (Cui et al., 2023), which contains both reward and judgment annotations produced by GPT4. Our preliminary experiments (Appendix A.6) show that CUT is not good using the original judgments in UltraFeedback. We find that the reason is that the judgments in UltraFeedback tend to commend the strengths of the given response. This type of judgment is unsuitable for our CUT, as we primarily use judgments for inappropriate token detection. Therefore, we re-collect judgments on the same instruction-response pairs from GPT4 using our prompt (Appendix A.2). Due to budget constraints, we randomly sample 4000 instructions (with 4 responses each, totaling 16000 pairs) for annotation. Implementation details are as follows:

- **DPO**: We formulate preference data by enumerating all possible pairs of responses to an instruction, excluding pairs with the same reward value. This results in 19956 examples for alignment.
- **CUT**: We use the 16000 instruction-response pairs but with our re-annotated judgments.

Results. Figure 4 shows that CUT can substantially surpass DPO by a large margin on AlpacaEval. It also shows that DPO performs not well on AlpacaEval when solely 1000 instructions are provided for alignment, indicating that reward-based DPO requires more training data than judgment-based CUT to achieve good alignment. The above observations suggest that judgments hold greater potential than rewards in aligning LLMs. CUT is comparable to or slightly worse than DPO on ARC, HellaSwag, MMLU, and TruthfulQA. We hypothesize that the performance discrepancy is partly caused by the evaluation protocols: the four tasks are ranking-based. As suggested Bansal et al. (2023), methods such as DPO, which leverage ranking data in the alignment possess inherent advantages in ranking-based tasks. We also provide a case study in Appendix A.7.

6 Conclusion

We systematically explored the alignment of LLMs through the lens of judgments. We investigated three potential methods that can be adapted for aligning LLMs with judgments but found them unable to fully capitalize on judgments. We proposed a novel framework CUT, that enables direct and explicit learning from judgments and facilitates fine-grained inappropriate content detection and correction. Extensive evaluations demonstrated the effectiveness of our CUT in various settings, including offline and online, specialist and generalist, as well as cold-start and warm-start scenarios. For example, the online alignment experiments showed that CUT can iteratively improve LLMs with up-to-date model-specific judgments, akin to how humans progressively refine their behaviors through ongoing feedback. Our analysis comparing rewards and judgments suggested that aligning LLMs with judgments is a promising research area.

Limitations

Quality of Judgment Models Despite the positive alignment results of our AI judge mentioned in Figure 3, we find the quality of its generated judgments is not satisfactory and significantly inferior to those generated by GPT4. Therefore, we discuss from the point of judgment generation and identify two limitations when interacting with AI judges:

- AI judges often make inaccurate judgments, leading to potential misclassification of inappropriate tokens as appropriate and vice versa. This may increase the risk of hallucination. To address this issue, periodically involving human annotators to provide accurate judgments can be a good attempt to reduce the hallucinations accumulated during interactions with AI judges.
- In an attempt to augment the training size, we incorporated the 1317 judgment data from Shepherd for training the AI judge. However, after including Shepherd, the AI judge's performance deteriorated, resulting in more illogical judgments such as "The original answer 100 is incorrect. The correct answer should be 100." We hypothesize that reasoning and math tasks from Shepherd are too complex for a 13b model to comprehend. Consequently, larger language models may be required to achieve better judgment generation quality, a notion supported by (Saunders et al., 2022).

Size of Alignment Data Due to budgetary constraints, our research currently involves experiments utilizing several thousands of judgment data. In future research endeavors, we would like to investigate the scaling law with an expanded volume of judgment data.

References

Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proc. of ACL*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *ArXiv preprint*, abs/2212.08073.

Hritik Bansal, John Dang, and Aditya Grover. 2023. Peering through preferences: Unraveling feedback acquisition for aligning large language models. *ArXiv preprint*, abs/2308.15812.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *ArXiv preprint*, abs/2310.01377.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *ArXiv preprint*, abs/2304.06767.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *ArXiv preprint*, abs/2308.07317.

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Dialogue learning with human-in-the-loop. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative judge for evaluating alignment. *ArXiv preprint*, abs/2310.05470.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023a. Languages are rewards: Hindsight finetuning using human feedback. *ArXiv preprint*, abs/2302.02676.

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

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *ArXiv preprint*, abs/2303.17651.

Deepak Nathani, David Wang, Liangming Pan, and William Yang Wang. 2023. Maf: Multi-aspect feedback for improving reasoning in large language models. *ArXiv preprint*, abs/2310.12426.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023a. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *ArXiv preprint*, abs/2302.12813.

Baolin Peng, Linfeng Song, Ye Tian, Lifeng Jin, Haitao Mi, and Dong Yu. 2023b. Stabilizing rlhf through advantage model and selective rehearsal. *ArXiv preprint*, abs/2309.10202.

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*.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *The Eleventh International Conference on Learning Representations*.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *ArXiv preprint*, abs/2206.05802.

Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Training language models with natural language feedback. *ArXiv preprint*, abs/2204.14146.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *ArXiv preprint*, abs/2303.16755.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv preprint*, abs/1707.06347.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. *ArXiv preprint*, abs/2306.17492.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

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

Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023a. Shepherd: A critic for language model generation. *ArXiv preprint*, abs/2308.04592.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023b. Aligning large language models with human: A survey. *ArXiv preprint*, abs/2307.12966.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. Generating sequences by learning to self-correct. *ArXiv preprint*, abs/2211.00053.

Jason Weston. 2016. Dialog-based language learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. 2023. Rlcd: Reinforcement learning from contrast distillation for language model alignment. *ArXiv preprint*, abs/2307.12950.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Tianshu Yu, Ting-En Lin, Yuchuan Wu, Min Yang, Fei Huang, and Yongbin Li. 2023. Constructive large language models alignment with diverse feedback. *ArXiv preprint*, abs/2310.06450.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *ArXiv preprint*, abs/2304.05302.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Tianjun Zhang, Fangchen Liu, Justin Wong, Pieter Abbeel, and Joseph E Gonzalez. 2023. The wisdom of hindsight makes language models better instruction followers. *ArXiv preprint*, abs/2302.05206.

Rui Zheng, Shihan Dou, Songyang Gao, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Limao Xiong, Lu Chen, et al. 2023. Secrets of rlhf in large language models part i: Ppo. *ArXiv preprint*, abs/2307.04964.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *ArXiv preprint*, abs/2305.11206.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *ArXiv preprint*, abs/1909.08593.

A Appendix

A.1 Implementations

We train our models using LoRA (Hu et al., 2022) and follow the best configurations suggested by Platypus (Lee et al., 2023). The tradeoff hyperparameter λ is selected from $\{1.1, 1.2, 1.5\}$ and the unlikelihood weight α is selected from $\{0.25, 0.5, 0.75, 1\}$. We adopt the Alpaca template (Taori et al., 2023) for fine-tuning and inference. Figure 5 shows the templates when we apply CUT to align LLMs. Figure 6 shows the inference template, which does not necessitate judgments.

A.2 Prompt for Judgment Annotation

Figure 8 illustrates the prompt employed to request GPT-4's assistance in annotating judgments. We consider the judgment that begins with the keyword "Perfect." to be a positive judgment; otherwise, it is deemed a negative judgment. GPT-4 demonstrates proficiency in fulfilling this requirement. Figure 9 shows the template used for training AI judges.

A.3 Offline Alignment Tasks

We conduct experiments on two tasks, a general instruction-following task, and a specific NLP task (summarization):

- **General Instruction-following:** We train models on the Shepherd dataset (Wang et al., 2023a), which consists of judgment data on diverse NLP tasks such as math word problems and commonsense reasoning. There are 1317 examples in total. For evaluation, we report model performance on four ranking-based and one generation-based LLM benchmarks, where ranking-based evaluation tests an LLM's ability to *select* the best response from a set of candidate responses, while generation-based evaluation assesses an LLM's ability to *generate* high-quality responses. Following the Open LLM Leaderboard (Gao et al., 2021), the ranking-based benchmarks are 25-shot ARC (Clark et al., 2018), 10-shot HellaSwag (Zellers et al., 2019), 5-shot MMLU (Hendrycks et al., 2021), and 0-shot TruthfulQA (Lin et al., 2022). The generation-based benchmark is AlpacaEval[1].
- **Summarization:** We use the summarization dataset with judgment annotations produced by (Saunders et al., 2022). We use the training split (10827 examples) to train our models and report ROUGE scores (Lin, 2004) on the test split (1939 examples).

[1]Following conventions, GPT4 is utilized to judge the winning rate of the responses generated by our models against those produced by DaVinci003.

A.4 Downsampling Align-P

As LLaMA2-chat has already undergone extensive alignment training, its responses to the Stanford Alpaca instructions are generally of high quality. In fact, 713 out of 1000 responses generated by LLaMA2-chat receive positive judgments, resulting in a substantial proportion of Align-P examples. To investigate the effect of the proportion of Align-P examples, we undertake a downsampling process for these examples. The performance of various downsampling ratios is illustrated in Figure 7. Our findings indicate that maintaining a moderate percentage of Align-P examples is crucial. We conjecture that preserving a certain number of Align-P examples allows the model to sustain its capacity to generate satisfactory responses, while too many Align-P examples may lead to overfitting, thereby disrupting the alignment process. In subsequent experiments, we keep a ratio of 0.25.

A.5 Case Study: Online Alignment

Table 6 presents three examples of model-generated responses after each training iteration. In general, the responses produced by different models do not display significant variations, as most content is satisfactory even before training and kept unchanged in subsequent iterations. Meanwhile, the generation quality exhibits a gradual improvement, characterized by the correction of specific errors and the inclusion of valuable improvements.

- **Case 1:** CUT 3+ introduces a crucial constraint that influences the color of the sky.
- **Case 2:** CUT 1+ amends a hallucination present in LLaMA2-chat's response (the fabricated file name "*First document.tex*"), though it introduces an additional mistake elsewhere. Fortunately, CUT 4+ is capable of rectifying the newly introduced error and providing a concise and satisfactory response.
- **Case 3:** CUT 1+/2+/3+ adds a sentence that closely resembles the style of a Twitter post. Moreover, CUT 4+ incorporates hashtags, further enhancing the resemblance to the typical format of a Twitter post.

Instruction:	What color is the sky?
LLaMA2-chat:	The color of the sky is blue.
CUT 1+:	The color of the sky is blue.
CUT 2+:	The color of the sky is blue.
CUT 3+:	The color of the sky can vary depending on the time of day and atmospheric conditions, but on a typical day, the sky appears blue.
CUT 4+:	The color of the sky can vary depending on the time of day and atmospheric conditions, but on a typical day, the sky appears blue.
Instruction:	What is the output of the given LATEX code? \documentclass{article} \begin{document} First document. This is a simple example, with no extra parameters or packages included. \end{document}
LLaMA2-chat:	The output of the given LATEX code is a document classified as an "article" with the file name "First document.tex". The document contains only one line of text "First document. This is a simple example, with no extra parameters or packages included.
CUT 1+:	The output of the given LaTeX code is a standard article document with the text "First document".
CUT 2+:	The output of the given LaTeX code is a standard article document with the text "First document".
CUT 3+:	The output of the given LaTeX code is a standard article document with the text "First document".
CUT 4+:	Sure! Here's the output of the given LaTeX code: First document. This is a simple example, with no extra parameters or packages included.
Instruction:	Write a good Twitter Bio. Try to cover all the provided main points in your generated bio. Name: Jane Main points: - Architect - Interested in Tech Written in first person Use casual tone
LLaMA2-chat:	Hey there! I'm Jane, a passionate architect with a love for all things tech. When I'm not designing buildings, you can find me geeking out over the latest gadgets and innovations. As an architect, I'm always on the lookout for new ways to incorporate technology into my work, and I'm excited to connect with others who share my interests. Let's build something amazing together! #tech #architecture #innovation
CUT 1+:	Hey there, I'm Jane! I'm an architect with a passion for all things tech. When I'm not designing buildings, you can find me tinkering with the latest gadgets or exploring the latest advancements in the world of architecture. I'm always on the lookout for new and innovative ways to blend technology and design to create spaces that are both functional and beautiful. Follow me for a mix of architecture, tech, and design inspiration!
CUT 2+:	Hey there, I'm Jane! I'm an architect with a passion for all things tech. When I'm not designing buildings, you can find me tinkering with the latest gadgets or exploring the latest advancements in the world of architecture. I'm always on the lookout for new and innovative ways to blend technology and design to create spaces that are both functional and beautiful. Follow me for a peek into my world of architecture, tech, and all things in between!
CUT 3+:	Hey there, I'm Jane! I'm an architect with a passion for all things tech. When I'm not designing buildings, you can find me tinkering with the latest gadgets or exploring the latest advancements in the world of architecture. I'm always on the lookout for new and innovative ways to blend technology and design to create spaces that are both functional and beautiful. Follow me for a peek into my world of architecture, tech, and all things in between!
CUT 4+:	Hey there, I'm Jane! I'm an architect with a passion for all things tech. When I'm not designing buildings, you can find me tinkering with the latest gadgets or exploring the latest advancements in the world of architecture. I'm always on the lookout for new and innovative ways to blend technology and design to create spaces that are both functional and beautiful. Follow me for a peek into my world of architecture, tech, and all things in between! #architect #tech #innovation #design

Table 6: Case study for online iterative alignment. Some satisfactory and unsatisfactory text segments are labeled in red and blue respectively.

Align-P

Below is an instruction that describes a task. Write a response that appropriately completes the request.

Instruction:
{instruction}

Response:
{satisfactory response}

Align-N

Below is an instruction that describes a task. Write a response to the instruction and the response should match the corresponding judgment.

Instruction:
{instruction}

Judgment:
{negative judgment}

Response:
{unsatisfactory response}

Misalign

Below is an instruction that describes a task. Write a response that appropriately completes the request.

Instruction:
{instruction}

Response:
{unsatisfactory response}

Figure 5: The template used for aligning LLMs through CUT.

Inference

Below is an instruction that describes a task. Write a response that appropriately completes the request.

Instruction:
{instruction}

Response:

Figure 6: The inference template.

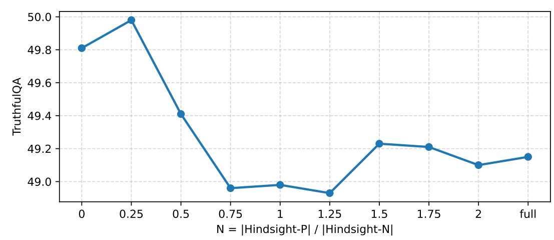


Figure 7: The effect of Align-P examples during online iteration.

A.6 Analysis of UltraFeedback's Judgments

Qualitative Analysis Table 7 shows that judgments in UltraFeedback tend to commend the strengths of given responses. When implementing our CUT method with such judgments, it remains uncertain whether the tokens detected by CUT are more relevant to the strengths or weaknesses of the response. Consequently, there is a potential risk of inaccurately categorizing appropriately generated tokens as inappropriate ones.

Quantitative Analysis We proceed with our investigation by performing a quantitative analysis, employing the CUT method utilizing judgments from UltraFeedback, as well as our re-annotated judgments. In this analysis, we use 1000 instructions, each associated with 4 responses, resulting in a total of 4000 instruction-response pairs.

The implementation details are as follows:

- **CUT-UF**: We fine-tune the base model on above 4000 instruction-response pairs and their original judgments from UltraFeedback using CUT.
- **CUT**: We use the same 4000 instruction-response pairs but with our re-annotated judgments.

As shown in Table 8, CUT consistently improves over CUT-UF on all five tasks for two base models. These results support our claims that judgments based solely on weaknesses are more suitable than those involving strength in the implementation of CUT to align LLMs.

A.7 Case Study: CUT v.s. DPO

As shown in Table 9, for a qualitative comparison of DPO and CUT, we perform a close examination of the generated responses from two methods. We find that DPO's responses are more polite. However, CUT's responses often exhibit greater specificity (Case 1), offer more helpful information (Case 2), and adhere more closely to the given instruction (Case 3), compared to those produced by DPO.

GPT4 Judgment Annotation

System content:
Below is an instruction that describes a task and a potential response. Evaluate the response and provide valuable judgments to the response. If the response is perfect, please only reply with 'perfect'. Otherwise, please indicate precisely what mistakes it has.

User content:
Instruction:
{instruction}

Response:
{response}

Figure 8: The prompt for asking GPT4 in annotating judgment.

Training Template for AI Judges

Below is an instruction-response pair. Write a judgment to evaluate the quality of this response. Then reply with 'Yes.' or 'No.' to show your decision on whether the response is perfect.

Instruction:
{instruction}

Response:
{response}

Judgment:
{judgment}
{decision}

Figure 9: The template used for training AI judges.

	CUT	UltraFeedback
Instruction	Tell me the name of the capital of France. Output the name.	Given a list of items, indicate which items are difficult to recycle. Cans, Plastic bags, Glass bottles
Response	Sure thing! The capital of France is Paris.	Plastic bags are difficult to recycle because they often get tangled in recycling machinery and contaminate other recyclables.
Judgment	The response is incorrect. The instruction asked for the output to be just the name of the capital, but the assistant provided additional information. The correct response should be simply "Paris“.	Your answer correctly identified plastic bags as difficult to recycle and provided a good reason why. However, you missed addressing the recyclability of cans and glass bottles, which were also part of the instruction. Make sure to cover all points mentioned in the instruction for a comprehensive answer.

Table 7: Judgment examples from UltraFeedback and our CUT work. The strengths of the response in the UltraFeedback judgment are labeled in red.

Method	ARC	HellaSwag	MMLU	TruthfulQA	AlpacaEval
LLaMA2	59.72	**81.39**	54.97	36.28	1.87
+ CUT-UF	60.32	80.42	54.37	45.06	48.88
+ CUT	**61.69**	81.00	**55.22**	**47.48**	**65.30**
LLaMA2-chat	58.02	79.89	54.52	45.44	81.09
+ CUT-UF	58.70	77.92	54.30	50.48	83.19
+ CUT	**59.13**	79.48	**54.86**	**53.18**	**86.36**

Table 8: Comparison between judgments of UltraFeedback and our re-annotated judgments.

Instruction:	How did US states get their names?
DPO:	Good question! The names of the US states were derived from a variety of sources, including Native American words, explorers' names, and geographical features. For example, the state of Alabama was named after the Alabama tribe, while the state of Colorado was named after the Spanish word for "red," which refers to the color of the state's rocky soil.
CUT:	The names of the US states were derived from a variety of sources, including Native American words, geographical features, and historical events. Here are some examples of how some of the states got their names: 1. Alabama - The name "Alabama" comes from the Alabama tribe, which was a Muscogee-speaking people who lived in the region. 2. Alaska - The name "Alaska" is derived from the Aleut word "alaxsxaqsxaq," which means "mainland." ... 49. Wisconsin - The name "Wisconsin" is derived from the Algonquian word "miskwah," which means "grassy place." 50. Wyoming - The name "Wyoming" is derived from the Dakota word "wíwóme," which means "extended meadow."
Instruction:	I'm trying to teach myself to have nicer handwriting. Can you help?
DPO:	Of course! I'd be happy to help you improve your handwriting. What specific areas are you looking to work on? Do you want to focus on letter formation, spacing, or overall legibility?
CUT:	Of course! Improving your handwriting can be a fun and rewarding process. Here are some tips to help you achieve nicer handwriting: 1. Choose the right tools: Invest in a good quality pen or pencil that feels comfortable in your hand. Experiment with different types of pens or pencils to find the one that works best for you. 2. Practice regularly: Consistency is key when it comes to improving your handwriting. Set aside some time each day to practice writing. Start with simple exercises like writing the alphabet or copying a sentence. ... 10. Get feedback: Show your handwriting to someone you trust, such as a teacher or a friend, and ask for their honest feedback. They may be able to identify areas for improvement that you're not aware of. Remember, improving your handwriting takes time and practice. Be patient and persistent, and you'll see improvement over time. Good luck!
Instruction:	Write "Test"
DPO:	Sure! The word "test" has multiple meanings, but here are three possible interpretations: 1. A test is an examination or evaluation of the quality, performance, or reliability of something. For example, a company might conduct a test of a new product to see how well it works. 2. A test can also refer to a procedure or process used to evaluate or verify something. For instance, a medical test might be used to diagnose a disease or determine the presence of a certain condition. 3. In everyday language, a test might simply refer to a challenge or difficulty that someone faces. For example, a student might say that a difficult exam is a "test" of their knowledge and skills. Is there anything else you'd like to know about the word "test"?
CUT:	Test

Table 9: Examples of responses generated by DPO and CUT respectively.