Aligning Language Models Using Follow-up Likelihood as Reward Signal

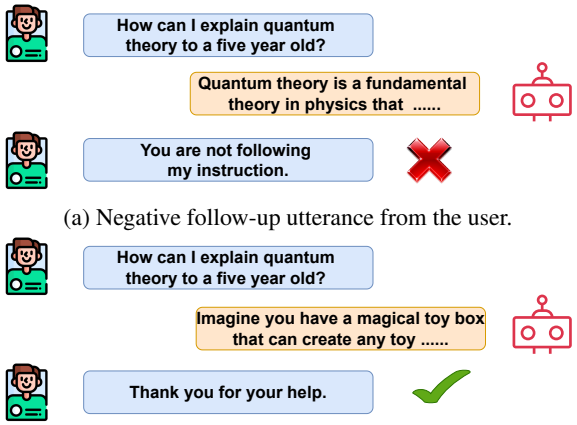
Chen Zhang[1], **Dading Chong**[2], **Feng Jiang**[3,4,5*]
Chengguang Tang[6], **Anningzhe Gao**[4], **Guohua Tang**[6], **Haizhou Li**[1,3,4]
[1]National University of Singapore, Singapore
[2]Peking University, China
[3]The Chinese University of Hong Kong, Shenzhen, China
[4]Shenzhen Research Institute of Big Data, China
[5]University of Science and Technology of China, China
[6]Tencent AI Lab, China
chen_zhang@u.nus.edu, jeffreyjiang@cuhk.edu.cn

Abstract

In natural human-to-human conversations, participants often receive feedback signals from one another based on their follow-up reactions. These reactions can include verbal responses, facial expressions, changes in emotional state, and other non-verbal cues. Similarly, in human-machine interactions, the machine can leverage the user's follow-up utterances as feedback signals to assess whether it has appropriately addressed the user's request. Therefore, we propose using the likelihood of follow-up utterances as rewards to differentiate preferred responses from less favored ones, without relying on human or commercial LLM-based preference annotations. Our proposed reward mechanism, "Follow-up Likelihood as Reward" (FLR), matches the performance of strong reward models trained on large-scale human or GPT-4 annotated data on 8 pairwise-preference and 4 rating-based benchmarks. Building upon the FLR mechanism, we propose to automatically mine preference data from the online generations of a base policy model. The preference data are subsequently used to boost the helpfulness of the base model through direct alignment from preference (DAP) methods, such as direct preference optimization (DPO). Lastly, we demonstrate that fine-tuning the language model that provides follow-up likelihood with natural language feedback significantly enhances FLR's performance on reward modeling benchmarks and effectiveness in aligning the base policy model's helpfulness.

1 Introduction

The recent development of large language models (LLMs) has revolutionized the field of natural language processing (Zhao et al. 2023a). Many chat-based LLMs are trained to align a large-scale pretrained language model to human preferences using techniques such as supervised fine-tuning (SFT) (Wei et al. 2022; Chung et al. 2022), reinforcement learning from human feedback (RLHF) (Christiano et al. 2017; Ziegler et al. 2019; Ouyang et al. 2022), or direct alignment from preferences (DAP) (Rafailov et al. 2023; Zhao et al. 2023b; Ethayarajh et al. 2024). All these techniques heavily rely on high-quality SFT or preference data, but collecting such data through human efforts is extremely expensive.



(a) Negative follow-up utterance from the user.

(b) Positive follow-up utterance from the user.

Figure 1: Followup Examples.

Lately, there has been a growing trend in the self-evolution of LLMs. This process allows the models to autonomously obtain, improve, and learn from their own generated experiences (Tao et al. 2024), reducing the need for costly human supervision. For instance, Yuan et al. (2024) introduces self-rewarding language models, starting with bootstrapping instructions from a base policy model to generate candidate responses. The model then employs "LLM-as-a-Judge" prompting (Zheng et al. 2023a) to evaluate its own outputs, creating a collection of self-curated preference data. The data are then used to align the language model through directed preference optimization (Rafailov et al. 2023). In a similar vein, many recent works (Xu et al. 2023c; Gulcehre et al. 2023; Chen et al. 2024b; Guo et al. 2024; Li et al. 2024c; Zhang et al. 2024b, inter alia) explore using online generations of language models for self-improvement in an iterative manner. At the core of these works is an automated procedure for response quality annotation, which uti-

*Corresponding Author

lizes either self or external reward signals to rank or score the response candidates (Chen et al. 2024a). The reliability of this procedure depends significantly on the reward model's language understanding and quality discrimination capabilities, which highly correlate with the model's scale and the quality of its training data (Zhang et al. 2024a). For example, GPT-4 (OpenAI 2023) emerges as an ideal annotator due to its large model size and high-quality instruction-tuning data.

However, training a reward model similar to GPT-4 or using API services to prompt such a model is resource-intensive. Therefore, finding alternative indicators of response quality is necessary. Drawing inspiration from natural human-human interactions where speakers continuously adapt their speech based on real-time feedback from their conversational partners, we postulate that the quality of the LLMs' responses may be implicitly derived from the real users' reactions, their follow-up utterances in particular. For instance, if an AI assistant is not following the user's instructions, the users are more likely to express disagreement or criticism than compliments or gratitude. This is illustrated by the examples in Figure 1.

Considering that it is infeasible to conduct large-scale quality annotation with an interactive setup between a real user and the language model, where the user provides real-time feedback to the language model generations, we approximate the user using an instruction-tuned LLM, which can either be the base policy model or another LM. This approach is motivated by the fact that existing instruction-tuned LLMs are fine-tuned using high-quality chat-style data (Chiang et al. 2023; Xu et al. 2023a) and hence, possess the knowledge of human conversational dynamics. Consequently, their likelihood of generating negative feedback for an appropriate response is lower than that of generating positive feedback, and vice versa.

Hence, we manually curate a set of positive and negative follow-up utterances. The conditional log probability of these follow-up utterances, given a prompt-response pair, serves as a reward signal to determine the helpfulness of the response. While prior works (Mehri and Eskenazi 2020; De Bruyn et al. 2022) utilize the likelihood of follow-up utterances as an automatic evaluation metric for open-domain dialogues, our work takes a significant leap beyond mere dialogue evaluation. We leverage follow-up utterances for automatic preference annotations of online generations of the policy model without the need for an external reward model or human efforts. Such an automatic annotation procedure simplifies the alignment process and opens the door to more autonomous and efficient LLM optimization.

In summary, we make the following contributions:

- We introduce a "Follow-up Likelihood as Reward" (FLR)[1] that automatically annotates preference data from base policy outputs. The annotated data is used for DAP fine-tuning, significantly enhancing the helpfulness of the base policy model. For example, Llama-3-8B-Instruct's length-controlled win rate is improved by 4.45% on Alpaca-Eval V2 after fine-tuning (§5.3).
- We demonstrate that FLR, without additional training, matches the performance of strong reward models, which are trained on large-scale human-annotated or GPT-4 annotated preference data, across 8 pairwise preference and 4 rating-based benchmarks (§5.1).
- Lastly, we demonstrate that fine-tuning the LM used in FLR with natural language feedback data significantly enhances the performance on the reward modeling task (§5.1) and results in better helpfulness alignment of the base policy LLM, compared to using the original FLR without fine-tuning (§5.3).

[1]Repository available at https://github.com/e0397123/FLR.

2 Related Work

Reward Model Reward models play a pivotal role in the LLM alignment pipeline, serving as proxies for human judgment to guide policy models in distinguishing between desirable and undesirable outputs. Lambert et al. (2024) categorizes reward models into three primary types: 1) classifier-based, 2) DPO-based, and 3) prompting or generation-based. The first category, exemplified by models such as Starling-RM-7B-alpha (Zhu et al. 2023), Oasst RM, and ArmoRM-Llama3-8B-v0.1 (Wang et al. 2024a), involves training a sequence classifier on top of a language model using human-annotated preference data. The second category, including Tulu-2 (Wang et al. 2023) and Zephyr-7b-alpha (Tunstall et al. 2023), directly computes rewards based on the conditional log probability of the response. The third category, such as auto-J (Li et al. 2024a), Prometheus (Kim et al. 2024b), and GPT-based LLMs (OpenAI 2023), uses a prompting approach to generate reward scores, following the LLM-as-a-Judge (Zheng et al. 2023a) framework. The first category demands additional training efforts on large-scale human-annotated data to achieve satisfactory performance, while the third category typically requires a powerful, large-scale instruction-following LLM to attain similar levels of performance. Training such an LLM is exceedingly costly. Experiments in both Lambert et al. (2024) and ours demonstrate the performance of the second category is sub-optimal compared to the first due to the lack of additional training.

However, FLR differs from all three categories. It shares the advantage of the second category by not requiring additional training on human-annotated data, yet it achieves performance comparable to that of the first category. Additionally, FLR paves the way for future research into leveraging feedback signals from real users in reward modeling.

LLM Alignment Post-training alignment of large language models has become a key focus in LLM research. Ouyang et al. (2022) demonstrated the efficacy of the RLHF pipeline in enhancing GPT-3's ability to follow instructions. Subsequent studies have explored alternatives to costly reinforcement training by proposing techniques for direct alignment from preferences (Rafailov et al. 2023; Ethayarajh et al. 2024; Zhao et al. 2023b). Additionally, there has been significant interest in the self-improvement capabilities of LLMs (Yuan et al. 2024; Guo et al. 2024; Wu et al. 2024b) leveraging the DAP techniques and preference annotations on the online generation of the policy model. Our work focuses on reward modeling and the automatic annotation of

preference data, making it applicable to all these algorithms. We also demonstrate the superiority of FLR in automatic preference annotation.

Recent studies have also explored the use of natural language feedback, produced by either humans or large language models (LLMs), to enhance the interpretability of LLM evaluations (Cui et al. 2023; Madaan et al. 2023; Wu et al. 2024a). This approach not only improves their utility but also aids in enhancing their helpfulness. Unlike existing methods, our work transforms this feedback into real-user follow-ups and utilizes it to refine the follow-up likelihood estimation in LLMs, thereby advancing the reward modeling capabilities of the proposed FLR mechanism.

3 Methodology

In this section, we begin by formally defining the tasks. Next, we describe the "Follow-up Likelihood as Reward (FLR)" procedure. Finally, we outline the automatic annotation process for the base language model's generations and explain how the data are used to align the base model.

3.1 Task Formulation

Rating-Based Quality Annotation Given a chat-style instruction-tuning dataset $\{p_i, r_i\} \in \mathcal{D}$, p_i is chat history between a user and a policy model and r_i is the model completion. p_i can be a single-turn instruction prompt from the user or a multi-turn interaction between the user and the policy model up to the most recent user turn. Based on certain criteria, we need to score r_i. Denote the score as s_{r_i}. In this paper, we focus on the helpfulness criteria of the completion as defined in Askell et al. (2021). We adopt an instruction-tuned LLM, denoted as $\mathcal{M}$ to conduct the FLR scoring mechanism. The effectiveness of FLR can be quantified by the agreement between $\{s_{r_i}\}_{i=1}^{|\mathcal{D}|}$ and the corresponding ground-truth ratings, $\{g_{r_i}\}_{i=1}^{|\mathcal{D}|}$. Common agreement metrics include Pearson, Spearman, and Kendall correlation coefficients. In most open-source instruction-tuning benchmarks (Ye et al. 2024; Kim et al. 2024a; Wang et al. 2024b), the ground-truth ratings are typically obtained via human evaluation or API-based prompting of GPT-4.

Preference Data Annotation Most formulations of preference annotations is similar to rating-based quality annotation. The difference is that instead of a single completion r_i, there are $k \geq 2$ candidates to score. The preference dataset, $\mathcal{D}$ consists of $\{p_i, r_i^1, r_i^2, \ldots, r_i^k\}$. Our goal is to rank $\{r_i^1, r_i^2, \ldots, r_i^k\}$ according to their extent of helpfulness. In this paper, we primarily study the special case of $k = 2$ and the effectiveness of the FLR scoring mechanism is quantified by the accuracy of scoring human preferred completion, r_i^+, higher than the disfavored completion, r_i^-, i.e., $s_{r_i^+} > s_{r_i^-}$.

3.2 Follow-up Likelihood as Reward (FLR)

Instead of directly prompting LLMs (Zheng et al. 2023a) or explicitly training a reward model, we leverage natural follow-up utterances as implicit feedback to reflect the helpfulness of r_i. This approach is grounded in the observation that many powerful instruction-tuned LLMs (Dubey et al. 2024; Yang et al. 2024) are fine-tuned on high-quality conversational data that inherently reflect human-like feedback dynamics, whereby helpful responses are more likely to be followed by positive feedback, while unhelpful responses are more likely to trigger negative reactions. This procedure does not require additional preference data collection or the training of a separate reward model. It relies on the language-understanding capability of the LLMs.

Let us define a positive follow-up utterance as f_j^+ and a negative follow-up utterance as f_j^-. The log probabilities of the instruction-tuned LLM, $\mathcal{M}$, generating f_j^+ and f_j^- conditioned on $\{p_i, r_i\}$ are given by:

$$\log p_{\mathcal{M}}(f_j^* \mid p_i, r_i) = \sum_{t=1}^{T} \log p_{\mathcal{M}}(f_{j,t}^* \mid p_i, r_i, f_{j,<t}^*) \quad (1)$$

where f_j^* can be either f_j^+ or f_j^- and T represents the total number of tokens in f_j^*.

Due to the large space of potential positive and negative follow-ups of $\{p_i, r_i\}$, relying on a single follow-up can introduce bias. Therefore, it is essential to consider a diverse set of follow-up utterances to accurately assess response quality. However, conducting an exhaustive search for all contextually relevant follow-ups is intractable. As a solution, we manually curate a set of positive and negative follow-ups based on the decomposition of helpfulness criteria. We leave the automatic search for suitable follow-ups to future work. Since helpfulness is an abstract concept, we decompose it into three fine-grained sub-categories: understanding, engagingness, and instruction-following. This decomposition is also motivated by prior works (Wu et al. 2023; Wang et al. 2024b). For each category, we come up with 10 different positive and negative follow-ups respectively. Table 1 gives some examples and the full list is in Appendix A[2].

We define the set of positive follow-ups for a particular sub-category, t, as F_t^+ where $f_j^+ \in F_t^+$ and the set of negative follow-ups for a particular sub-category as F_t^- where $f_j^- \in F_t^-$. To score $\{p_i, r_i\}$ based on the follow-ups of a particular sub-category, we define the following reward:

$$s_{r_i}^t = \frac{\sum_{f_j^+ \in F_t^+} \log p_{\mathcal{M}}(f_j^+ \mid p_i, r_i)}{|F_t^+|} - \frac{\sum_{f_j^- \in F_t^-} \log p_{\mathcal{M}}(f_j^- \mid p_i, r_i)}{|F_t^-|} \quad (2)$$

The above reward signal is directly applied to solve the rating-based quality annotation task by average aggregation of scores across the three sub-categories[3].

[2]The full appendix of the paper can be found at https://arxiv.org/abs/2409.13948.

[3]T = {understanding, engagingness, instruction-following}

Category	Positive/Negative Follow-Up Example
Understanding	That makes perfect sense! (✓) That makes no sense! (✗)
Engagingness	This is very interesting. (✓) That's not very interesting. (✗)
Instruction-Following	You did a fantastic job following my instructions. (✓) You didn't adhere to my instructions. (✗)

Table 1: Examples of positive and negative follow-ups for each sub-category of helpfulness.

$$s_{r_i} = \frac{\sum_{t \in T} s^t_{r_i}}{|T|} \tag{3}$$

For the pairwise preference annotation task, an ideal $\mathcal{M}$ satisfies the following constraint: $s^t_{r_i^+} > s^t_{r_i^-} \Rightarrow s_{r_i^+} > s_{r_i^-}$.

It is worth noting that FLR can seamlessly integrate into the autonomous self-evolution pipeline of language models via online DAP procedures (details in §3.4). it does not rely on collecting offline human preference data or external model supervision but instead leverages the internal knowledge of strong instruction-tuned models.

3.3 Enhance FLR with Natural Feedback Data

To enhance the ability of $\mathcal{M}$ in assigning high likelihoods to positive follow-ups and low likelihoods to negative follow-ups for helpful responses, and vice versa for unhelpful responses, we finetune $\mathcal{M}$ using natural language feedback data using a chat template as shown below

User: Please ask me a question or assign me a task.

LLM: [insert here the instruction prompt, p_i]

User: [insert here the response, r_i]

LLM: [insert here the feedback to (p_i, r_i), which is denoted as l_i]

In the chat template, the roles of the LLM and user are switched. A feedback data example can be found in Appendix B. During the supervised fine-tuning of $\mathcal{M}$, the model is optimized to generate the natural language feedback, l_i. To curate training data, a common approach is to mine it from real user-bot interactions. However, this real-world data often contains significant noise and demands costly cleaning efforts. As an alternative, we rewrite feedback from third-party evaluators in an existing dataset, transforming it into naturally occurring utterances with a conversational, first-person tone. Specifically, we prompt GPT-3.5-Turbo to rewrite the feedback sentences in a first-person tone. This strategy aligns the feedback more closely with real-user follow-ups. The instruction template to prompt ChatGPT is presented in Appendix C.

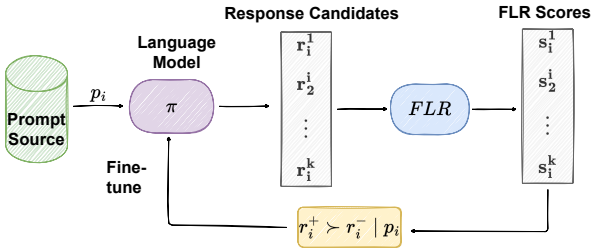


Figure 2: The procedure of aligning base LM. FLR denotes the "follow-up likelihood as reward" mechanism.

3.4 Aligning Base Language Model

Figure 2 presents the overall procedure of aligning a base policy model π. The main idea is to first sample instruction prompts p_i from a prompt source and π generates k response candidates, $\{r_i^1, r_i^2, \ldots, r_i^k\}$. Each candidate is assigned a reward score as defined in Eq.3. The candidate with the highest score is treated as the positive response r_i^+. The candidate with the lowest score is treated as the negative response r_i^-. This strategy helps avoid using similar pairs with slight differences in reward scores, which may confuse the generation model. Additionally, if the obtained pair contains responses with the same reward scores, it is filtered out. We experiment with DPO (Rafailov et al. 2023) and KTO (Ethayarajh et al. 2024) fine-tuning of p_i using the synthetic preference dataset constructed with the above procedure.

4 Experiment Setup

There are two parts of our experiments: (1) reward modeling and (2) helpfulness alignment. For (1), reward models are assessed on human-annotated pairwise preference and rating-based quality evaluation benchmarks (both tasks have been introduced in §2.1). (2) involves evaluating the helpfulness alignment of policy models after fine-tuning using the automatically curated preference data with FLR. §3.1 presents the training details. §3.2 introduces benchmarks and evaluation methodology used in both parts of the experiments. §3.3 describes the baselines for comparison.

4.1 Training Details

For $\mathcal{M}$, we experiment with Llama-3-8B-Instruct and Qwen2-7B-Instruct. We utilize the UltraFeedback (Cui et al.

Test Datasets	Size	Avg. #Prompt Words	Avg. #Turns per Prompt	Type
HH-RLHF Helpfulness (2022) (HH)	6,238	93.05	2.38	Pairwise
BeaverTails Helpfulness (2023) (BH)	2,985	13.17	1.00	Pairwise
SHP (2022)	18,409	148.79	1.00	Pairwise
Alpaca-Farm (2023) (AF)	17,701	28.57	1.00	Pairwise
MT-Bench Pairwise (2023a) (MP)	916	235.73	3.00	Pairwise
RewardBench Chat Easy (2024) (RCE)	358	41.92	1.00	Pairwise
RewardBench Chat Hard (2024) (RCH)	456	30.45	1.00	Pairwise
Preference Bench (2024a) (PB)	1,998	79.21	1.00	Pairwise
Feedback Bench (2024a) (FH)	1,000	79.25	1.00	Rating
FLASK (2024)	1,500	70.29	1.00	Rating
MT-Bench Rating (2023a) (MR)	1,000	141.97	2.05	Rating
HelpSteer (2024b)	1,789	432.42	1.00	Rating

Table 2: Statistics of the reward model benchmarks.

2023) dataset for rewriting, which is then employed to fine-tune $\mathcal{M}$. UltraFeedback contains 255,548 instances of (p_i, r_i, l_i) and we randomly sample 100K for fine-tuning.

For the helpfulness alignment experiment, we also adopt Llama-3-8B-Instruct and Qwen2-7B-Instruct as the base policy model π. LoRA (Hu et al. 2022) is applied to all fine-tuning experiments on a single NVIDIA A100 80GB GPU. The experimental settings for DPO and KTO fine-tuning follow the implementations from LLaMA-Factory (Zheng et al. 2024). We adopt the Nectar dataset (Zhu et al. 2023) as the prompt source. The 183K prompts in Nectar are a mixture of diverse sources, including lmsys-chat-1M (Zheng et al. 2023b), ShareGPT, Anthropic/HH-RLHF (Bai et al. 2022), UltraFeedback, Evol-Instruct (Xu et al. 2023b), and Flan (Longpre et al. 2023). For our experiments, we randomly sampled 100K instruction prompts from Nectar. Full details on reproducibitlity can be found in Appendix E.

4.2 Benchmarks and Evaluation

We assess the reward models using eight pairwise preference benchmarks and four rating-based single-response benchmarks, with their statistics in Table 2. Accuracy is reported for the pairwise benchmarks and Pearson correlation for the rating-based benchmarks. Additionally, we examine the reward models' ability to rank different LLMs by providing system-level correlations between the reward model scores and real-user ELO ratings from the LMSys Chatbot Arena (Chiang et al. 2024). We obtain the Arena dataset[4] from Lin et al. (2024) and it consists of 865 data instances per LLM for a total of 30 LLMs, including GPT-4-turbo (OpenAI 2023), Meta-Llama-3-70B-Instruct, and Mixtral-8x7B-Instruct (Jiang et al. 2024).

To evaluate FLR's contribution to helpfulness alignment, we use well-established benchmarks including Alpaca-Eval V2 (Li et al. 2023), WildBench V2 (Lin et al. 2024), and FLASK (Ye et al. 2024). Alpaca-Eval V2, WildBench V2, and FLASK contain 805, 1024, and 1700 instruction prompts respectively. WildBench V2 involves pairwise comparisons with the base policy model as the reference. For Alpaca-Eval V2, we report the length-controlled win rate against GPT-4 Preview (11/06) as per standard protocol, and the "alpaca_eval_gpt4_turbo_fn" annotator config is adopted. For FLASK, the GPT-4 evaluator is adopted to rate the quality of the model responses according to specific prompts. Furthermore, we assess the general understanding capability of the aligned models on the Open LLM benchmark. We follow the official implementations of the benchmarks and the fp16 model variants are used for inference.

[4]The ELO ratings are updated as of June 7, 2024.

4.3 Baselines

Our proposed FLR mechanism is compared to state-of-the-art reward models, including Prometheus-7b-v2.0 (Kim et al. 2024b), GPT-3.5-Turbo, Starling-RM-7B-alpha (Zhu et al. 2023), oasst-rm-2-pythia-6.9b-epoch-1, PairRM (Jiang, Ren, and Lin 2023), and ArmoRM-Llama3-8B-v0.1 (Wang et al. 2024a). It's important to note that these RMs are trained on a diverse mix of high-quality, human-annotated, or GPT-annotated preference data, whereas FLR operates without such annotations. As PairRM is trained only for preference annotations, its performance on rating-based benchmarks is not reported.

Moreover, FLR is compared to the direct scoring of responses based on their log probability, as judged by DPO-finetuned LMs. These include Tulu-2-dpo-7b (Wang et al. 2023), Qwen2-7B-Instruct (Yang et al. 2024), Zephyr-7b-alpha (Tunstall et al. 2023), and Llama-3-8B-Instruct. We follow the official implementation of RewardBench (Lambert et al. 2024) and prometheus-eval (Kim et al. 2024a).

For the alignment experiment, we primarily study how the alignment performance changes with respect to the base policy model. Note that we focus on models at the 7B scale due to their good balance between performance and computational resource requirements.

5 Experiment

This section presents the results and analysis of the reward modeling and alignment experiments.

Models	HH	BH	SHP	AF	MP	RCE	RCH	PB	AVG-P	MR	FLASK	FB	HelpSteer	AVG-R
Direct (Tulu-2-dpo-7b)	55.59	58.99	45.83	55.90	63.65	86.59	39.04	62.51	58.51	0.114	0.111	0.206	0.167	0.150
Direct (Qwen2-7B-Instruct)	54.15	52.86	40.81	54.77	63.43	86.03	47.37	54.50	56.74	0.184	0.117	0.050	0.077	0.107
Direct (Zephyr-7b-alpha)	53.01	54.17	40.49	56.29	61.35	80.17	53.73	61.41	57.58	0.191	0.004	0.201	0.057	0.113
Direct (Llama-3-8B-Instruct)	55.51	54.74	42.64	54.90	62.88	86.31	43.86	49.25	56.26	0.183	0.089	0.024	0.127	0.106
GPT-3.5-Turbo	59.03	63.87	61.17	59.70	71.89	82.26	43.09	84.18	65.65	0.439	0.016	0.023	0.025	0.126
Prometheus-7B	**62.95**	**66.03**	55.13	**64.01**	**80.35**	92.18	48.03	**95.45**	**70.52**	0.312	0.251	**0.871**	0.390	0.456
Oasst-Pythia-6.9B	62.18	60.67	**68.61**	56.21	72.93	91.9	37.72	77.73	65.99	0.284	0.137	0.526	0.329	0.319
PairRM	62.06	56.82	54.91	58.04	76.09	84.64	50.66	81.03	65.53	-	-	-	-	-
Starling-RM-7B-alpha	**63.11**	**71.69**	59.67	**60.20**	**78.38**	**94.13**	40.35	86.69	69.28	0.492	0.235	**0.762**	0.449	0.485
ArmoRM-Llama3-8B-v0.1	-	63.72	**68.87**	**60.04**	77.40	**97.21**	**76.54**	**90.59**	**76.34**	0.406	**0.348**	**0.778**	**0.452**	**0.496**
FLR (Llama-3-8B-Instruct)	58.66	55.71	57.79	56.07	74.56	90.50	63.16	75.03	66.44	**0.555**	0.285	0.509	0.383	0.433
FLR (Qwen2-7B-Instruct)	57.23	58.29	59.37	53.39	71.62	80.45	55.26	73.12	63.59	0.403	0.157	0.423	0.296	0.320
FLR-FT (Llama-3-8B-Instruct)	**62.92**	**65.09**	**61.19**	59.99	**79.91**	**95.53**	**55.92**	**87.64**	**71.02**	**0.606**	**0.430**	0.705	**0.566**	**0.577**
FLR-FT (Qwen2-7B-Instruct)	62.50	61.11	57.17	59.28	72.93	92.18	**69.30**	85.39	69.98	**0.571**	**0.504**	0.701	**0.524**	**0.575**

Table 3: Results on the pairwise preference and rating-based helpfulness benchmarks. AVG-P (%) and AVG-R denote the average scores of pairwise preference and rating-based benchmarks respectively. The HH score for ArmoRM is omitted as it is trained on the data, achieving nearly 100% accuracy. FT refers to fine-tuning with the natural language feedback. Top-3 scores for each benchmark are highlighted in bold. "Direct" denotes direct scoring of the response by the LM using the log probability.

5.1 Reward Modeling Results

Main Benchmark Results Table 3 presents the accuracy scores (%) and Pearson correlations of various reward models on 8 pairwise preference datasets and 4 rating-based helpfulness datasets respectively. First, we observe that FLR performs significantly better than directly using the response likelihood. For instance, the average pairwise preference accuracy of FLR (Llama-3-8B-Instruct) is 66.44%, compared to 56.26% for Direct (Llama-3-8B-Instruct), showing a roughly 10% difference. Additionally, the average Pearson correlation achieved by FLR (Llama-3-8B-Instruct) is 0.327 points higher than that of Direct (Llama-3-8B-Instruct). We can also observe that FLR (Llama-3-8B-Instruct) without fine-tuning outperforms GPT-3.5-Turbo, PairRM, and Oasst-rm-2-pythia-6.9b-epoch-1, three strong lightweight reward models. These observations demonstrate that using real user follow-ups as an indication of response quality is a promising direction of reward modeling, given the presence of a strong instruction-tuned LM. Even compared to Prometheus-7b-v2.0, Starling-RM-7B-alpha, and ArmoRM-Llama3-8B-v0.1, which are trained on large-scale and high-quality human-annotated or GPT-4-annotated preference data, the performance of FLR without fine-tuning looks promising, especially on the rating-based benchmarks.

After fine-tuning with the natural language feedback data, the performance of FLR improves significantly. FLR (Llama-3-8B-Instruct - FT) achieves an average accuracy gain of 4.58% on the pairwise preference datasets and an increase of 0.144 in average Pearson correlation on the rating-based benchmarks compared to FLR (Llama-3-8B-Instruct). Significant performance boost can also be observed when using Qwen2-7B-Instruct. Moreover, the FLR fine-tuned variants' reward modeling performance consistently ranks among the top three of all reward models on most of the benchmarks. On average, FLR (Llama-3-8B-Instruct - FT) ranks third on pairwise preference benchmarks and first on rating-based benchmarks.

It's important to note that while ArmoRM-Llama3-8B-v0.1 ranks first on pairwise preference benchmarks, it has been extensively trained on around 1 million annotated pairs from 10 diverse, high-quality human preference datasets. Similarly, Prometheus-7b-v2.0 has also been trained on 20K preference data and 20K single-rating data, both with high-quality annotations. In contrast, vanilla FLR operates without such data, and its fine-tuning with approximately 100K natural language feedback samples isn't directly optimized for pairwise preference reward modeling. Moreover, the feedback data can be curated from real user-bot interactions or evaluation logs from third-party evaluators, showcasing the potential of using more accessible and naturally occurring data sources for effective reward modeling, bypassing the need for costly human or commercial LLM annotations.

Models	Pearson	Spearman	Kendall Tau
Prometheus-7B	0.683	0.673	0.492
Oasst-Pythia-6.9B	0.755	0.702	0.533
Starling-RM-7B-alpha	0.849	0.850	0.658
ArmoRM-Llama3-8B-v0.1	**0.909**	**0.897**	**0.751**
FLR (Llama3-8B-Instruct)	**0.897**	**0.899**	**0.727**
FLR (Qwen2-7B-Instruct)	0.631	0.586	0.454
FLR-FT (Llama3-8B-Instruct)	0.860	0.895	**0.727**
FLR-FT (Qwen2-7B-Instruct)	**0.892**	**0.897**	**0.732**

Table 4: System-level correlations on Chatbot Arena.

Correlation with Chatbot Arena In addition to ranking or rating responses, reward models can efficiently evaluate and rank the performance of different models. This capability provides researchers with quick feedback on model performance, facilitating faster model development and iteration. We examine whether FLR benefits such a use case. Table 4 presents the system-level correlations of the reward models with real-user ELO ratings on LMSys Chatbot Arena. We can see that FLR (Llama-3-8B-Instruct), FLR-FT (Llama-3-8B-Instruct), and FLR-FT (Qwen2-7B-Instruct) achieve strong performance, comparable to the state-of-the-art reward model, ArmoRM-Llama3-8B-v0.1. The observations further reinforce the effectiveness of the FLR.

Models	Avg-P	Avg-R
FLR (Llama-3-8B-Instruct)	66.44	0.433
\ Engagingness	64.57	0.412
\ Understanding	66.27	0.405
\ Following	64.56	0.398
\ Positive Follow-ups	57.50	-0.118
\ Negative Follow-ups	54.73	0.169
FLR (Llama-3-8B-Instruct - FT)	71.02	0.577
\ Engagingness	70.22	0.573
\ Understanding	70.75	0.552
\ Following	69.92	0.512
\ Positive Follow-ups	56.27	-0.034
\ Negative Follow-ups	67.09	0.484

Table 5: Ablation study for the reward modeling task. The symbol "\" denotes exclusion.

5.2 FLR Ablation Study

As shown in Table 5, removing any of the three categories: engagingness, understanding, or instruction-following, leads to a performance reduction. This effect is observed in both Llama-3-8B-Instruct and Llama-3-8B-Instruct - FT. Removing the instruction-following category results in the most significant performance reduction, indicating that it is a crucial attribute of response helpfulness.

Additionally, we observe that using only positive follow-ups or only negative follow-ups significantly reduces performance. Using only positive follow-ups means retaining only the first term in Eq.2, while using only negative follow-ups involves leveraging only the second term in Eq.2. Ideally, FLR should maximize the first term and minimize the second term when evaluating a good response, whereas a poor response would exhibit the opposite characteristics. Our proposal of using the score differences between positive and negative follow-ups is more robust and demonstrates significantly better performance.

5.3 Alignment Performance

Main Results Table 6 presents the alignment performance of policy models fine-tuned with DPO or KTO using preference data annotated with our proposed FLR mechanism, compared to the un-finetuned models[5]. Remarkably, we can observe that Llama-3-8B-Instruct can self-improve using our FLR mechanism without external supervision. For example, Llama-3-8B-Instruct + FLR DPO achieves a 33.17% LC win rate on Alpaca-Eval V2, marking a 4.45% improvement. On WildBench V2, it records a 57.57% win rate over its base version. Meanwhile, Qwen2-7B-Instruct shows less significant self-improvement, with minor gains observed only on WildBench V2 and FLASK. This may be because FLR is sensitive to factors such as the training method and the type of training data used in the underlying language model. This insight further motivates enhancing FLR by fine-tuning the language model with natural language feedback data.

It can be observed that the fine-tuned LM delivers more accurate FLR rewards, i.e., better indication of response helpfulness. With FLR-FT DPO, both Llama-3-8B-Instruct and Qwen2-7B-Instruct demonstrate significant improvements across all benchmarks. Notably, Qwen2-7B-Instruct + FLR DPO achieves the highest LC win rate of 34.26% on Alpaca-Eval V2 and a rating of 3.95 out of 5 on FLASK.

We can also observe that DPO fine-tuning yields superior alignment performance compared to KTO, consistent with previous findings (Rasul et al. 2024) that DPO outperforms KTO under paired preference settings. However, KTO's flexibility lies in its ability to function without paired data, making it more versatile than DPO. It is worth noting that FLR is compatible with all the DAP algorithms.

General Understanding Capability We also evaluate whether fine-tuning policy models using preference data annotated with FLR enhances response helpfulness without compromising their general understanding capability. The average scores of different models on the Open LLM Leaderboard are presented in Table 6. The detailed results for each dataset in the leaderboard are presented in Table 13 of Appendix D. We observe that FLR does not compromise model performance on the Open LLM leaderboard; in fact, it enhances it. For instance, Llama-3-8B-Instruct + FLR DPO achieves an average score of 0.648 compared to 0.644 for the base model. FLR-FT DPO delivers the best performance, with scores of 0.652 and 0.657 for the respective Llama-3-8B-Instruct and Qwen2-7B-Instruct base models.

Comparative Analysis Table 7 and Table 8 present the results of various online DPO pipelines on Alpaca-Eval V2, Arena-Hard (Li et al. 2024b), and FLASK respectively. Table 7 also highlights the key differences among them. We apply the same experiment recipe across pipelines, including Llama-3-8B-Instruct as the base policy model, Lora as the fine-tuning approach, DPO as the alignment technique, and the same set of training hyperparameters (details shown in Appendix E). We observe a performance gap between the state-of-the-art reward model, ArmoRM-Llama3-8B-v0.1, and our proposed FLR mechanism. ArmoRM achieves a

Models	FLASK	AlpacaEval LC	WildBench	Open LLM
Llama-3-8B-Instruct	3.76	28.72	50.00	0.644
+ FLR DPO	3.72	33.17	57.57	0.648
+ FLR KTO	3.73	30.59	55.50	0.640
+ FLR-FT DPO	**3.83**	**34.22**	**63.62**	**0.652**
+ FLR-FT KTO	3.76	33.89	60.50	0.643
Qwen2-7B-Instruct	3.87	29.74	50.00	0.639
+ FLR DPO	3.92	28.78	55.57	0.632
+ FLR KTO	3.89	26.15	54.49	0.640
+ FLR-FT DPO	**3.95**	**34.26**	61.52	**0.657**
+ FLR-FT KTO	3.92	33.17	**63.93**	0.643

Table 6: Alignment performance. The last column is the average score on the Open LLM Leaderboard. LC refers to the length-controlled win rate. FLR-FT refers to using the fine-tuned Llama-3-8B-Instruct to generate follow-up likelihood while FLR refers to using the base LM without fine-tuning.

[5]FLASK evaluates 12 quality aspects of a response, we report the average score across all aspects here. Details can be found in Table 12 of Appendix D.

Methods	LC / Win Rate / Length	RM	#External Anno	Reward Signal	Anno Type	Prompt Source
PairRM	35.92 / 33.04 / 1847	PairRM	~490K	Human, GPT-4	Pairwise	UltraFeedback
ArmoRM	41.40 / 39.13 / 1904	ArmoRM	~1M	Human, GPT-4	Pairwise	UltraFeedback
FLR (UltraFeedback)	32.60 / 34.22 / 2078	FLR	0	Follow-ups	-	UltraFeedback
FLR-FT (UltraFeedback)	35.18 / 35.47 / 2003	FLR-FT	100K	Follow-ups	NL Feedback	UltraFeedback
FLR (Nectar)	33.17 / 35.16 / 2091	FLR	0	Follow-ups	-	Nectar-100K
FLR-FT (Nectar)	34.22 / 35.22 / 2049	FLR-FT	100K	Follow-ups	NL Feedback	Nectar-100K

Table 7: Comparative analysis on Alpaca-Eval V2. "LC", "FT", "Anno", and "NL" denote "length-controlled", "fine-tuned", "annotations", and "natural language" respectively. Results of the LLaMA-3-8B-Instruct base policy is provided for reference.

Methods	Arena-Hard	95% CI	FLASK
LLaMA-3-8B-Instruct	20.6	(-2.0, 1.9)	3.76
PairRM	25.2	(-2.1, 2.0)	3.81
ArmoRM	28.4	(-1.9, 1.9)	3.81
FLR (UltraFeedback)	22.2	(-2.0, 1.9)	3.76
FLR-FT (UltraFeedback)	24.5	(-1.7, 1.5)	3.83
FLR + (Nectar)	21.7	(-2.1, 2.4)	3.72
FLR-FT + (Nectar)	24.8	(-2.6, 2.3)	3.83

Table 8: Comparative analysis on Arena-Hard and FLASK.

41.40% LC win rate on Alpaca-Eval V2 and 28.4% win rate on Arena-Hard respectively, whereas the best-performing FLR variant achieves 35.18% and 24.8% on the two benchmarks respectively. The gap can be attributed to the significant differences in both the quantity and diversity of annotated data utilized. ArmoRM, a Mixture-of-Experts (MoE) model, is trained on approximately 1 million pairwise annotations from humans or GPT-4, optimized across 19 reward objectives. In contrast, FLR bypasses the need for annotated preference data by deriving reward signals from naturally occurring follow-ups.

Notably, FLR performs comparably to PairRM, which has been trained on approximately 490K high-quality annotations, on all three benchmarks. This underscores the potential of leveraging naturally occurring follow-ups as a reward signal. Building on FLR's promise, our results further demonstrate that fine-tuning with natural language (NL) feedback can enhance alignment performance between FLR and FLR-FT, though the increase in LC win rate remains modest. Future work could explore more effective ways to leverage NL feedback or focus on curating higher-quality NL feedback to further enhance FLR's performance.

Additionally, we also examined the impact of using different prompt sources in the alignment pipeline and found no significant difference between using UltraFeedback and Nectar, such as 32.60% vs 33.17% LC win rate for FLR on Alpaca-Eval, 22.2% vs 21.7% win rate on Arena-Hard, and 3.76 vs 3.72 on FLASK. The observation is potentially due to the similarity in the distribution of their data.

6 Conclusion

In conclusion, our proposed 'Follow-up Likelihood as Reward' (FLR) mechanism offers a novel approach to reward modeling by utilizing user follow-up utterances as feedback, without relying on human or LLM-based annotations. FLR matches the performance of existing strong reward models on multiple benchmarks and enables automatic mining of preference data to enhance model helpfulness through direct alignment methods. Furthermore, fine-tuning the language model with natural language feedback significantly enhances FLR's effectiveness in reward modeling and, in turn, improves the alignment of the base policy models.

Acknowledgments

This research is supported by the project of Shenzhen Science and Technology Research Fund (Fundamental Research Key Project Grant No. JCYJ20220818103001002), Shenzhen Science and Technology Program (Grant No. ZDSYS20230626091302006), Key Project of Shenzhen Higher Education Stability Support Program (Grant No. 2024SC0009), SRIBD Innovation Fund (Grant No. K00120240006), and Shenzhen Science and Technology Program (Grant No. RCBS20231211090538066).

References

Askell, A.; Bai, Y.; Chen, A.; Drain, D.; Ganguli, D.; Henighan, T.; et al. 2021. A General Language Assistant as a Laboratory for Alignment. *arXiv preprint arXiv: 2112.00861*.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; et al. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv: Arxiv-2204.05862*.

Chen, Y.; Zhang, C.; Luo, D.; D'Haro, L. F.; Tan, R.; and Li, H. 2024a. Unveiling the Achilles' Heel of NLG Evaluators: A Unified Adversarial Framework Driven by Large Language Models. In *Findings of ACL 2024*.

Chen, Z.; Deng, Y.; Yuan, H.; Ji, K.; and Gu, Q. 2024b. Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models. *arXiv preprint arXiv: 2401.01335*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Chiang, W.-L.; Zheng, L.; Sheng, Y.; Angelopoulos, A. N.; Li, T.; Li, D.; Zhang, H.; Zhu, B.; Jordan, M.; Gonzalez, J. E.; and Stoica, I. 2024. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. arXiv:2403.04132.

Christiano, P. F.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Proc. of NeurIPS*, 4299–4307.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling instruction-finetuned language models. *ArXiv preprint*, abs/2210.11416.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; Zhu, W.; Ni, Y.; Xie, G.; Liu, Z.; and Sun, M. 2023. UltraFeedback: Boosting Language Models with High-quality Feedback. arXiv:2310.01377.

De Bruyn, M.; Lotfi, E.; Buhmann, J.; and Daelemans, W. 2022. Open-Domain Dialog Evaluation Using Follow-Ups Likelihood. In Calzolari, N.; Huang, C.-R.; Kim, H.; Pustejovsky, J.; Wanner, L.; Choi, K.-S.; Ryu, P.-M.; Chen, H.-H.; Donatelli, L.; Ji, H.; Kurohashi, S.; Paggio, P.; Xue, N.; Kim, S.; Hahm, Y.; He, Z.; Lee, T. K.; Santus, E.; Bond, F.; and Na, S.-H., eds., *Proc. of COLING*, 496–504.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv: 2407.21783*.

Dubois, Y.; Li, C. X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Proc. of NeurIPS*.

Ethayarajh, K.; Choi, Y.; and Swayamdipta, S. 2022. Understanding Dataset Difficulty with $\mathcal{V}$-Usable Information. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proc. of ICML*, volume 162 of *Proceedings of Machine Learning Research*, 5988–6008.

Ethayarajh, K.; Xu, W.; Muennighoff, N.; Jurafsky, D.; and Kiela, D. 2024. KTO: Model Alignment as Prospect Theoretic Optimization. *arXiv preprint arXiv: 2402.01306*.

Gulcehre, C.; Paine, T. L.; Srinivasan, S.; Konyushkova, K.; Weerts, L.; Sharma, A.; Siddhant, A.; Ahern, A.; Wang, M.; Gu, C.; Macherey, W.; Doucet, A.; Firat, O.; and de Freitas, N. 2023. Reinforced Self-Training (ReST) for Language Modeling. *arXiv preprint arXiv: 2308.08998*.

Guo, S.; Zhang, B.; Liu, T.; Liu, T.; Khalman, M.; Llinares, F.; Rame, A.; Mesnard, T.; Zhao, Y.; Piot, B.; et al. 2024. Direct Language Model Alignment from Online AI Feedback. *ArXiv preprint*, abs/2402.04792.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proc. of ICLR*.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Chen, B.; Sun, R.; Wang, Y.; and Yang, Y. 2023. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Proc. of NeurIPS*.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; et al. 2024. Mixtral of Experts. *arXiv preprint arXiv: 2401.04088*.

Jiang, D.; Ren, X.; and Lin, B. Y. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proc. of ACL*, 14165–14178.

Kim, S.; Shin, J.; Cho, Y.; Jang, J.; Longpre, S.; Lee, H.; Yun, S.; Shin, S.; Kim, S.; Thorne, J.; and Seo, M. 2024a. Prometheus: Inducing Fine-Grained Evaluation Capability in Language Models. In *Proc. of ICLR*.

Kim, S.; Suk, J.; Longpre, S.; Lin, B. Y.; Shin, J.; Welleck, S.; Neubig, G.; Lee, M.; Lee, K.; and Seo, M. 2024b. Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models. *arXiv preprint arXiv: 2405.01535*.

Lambert, N.; Pyatkin, V.; Morrison, J.; Miranda, L.; Lin, B. Y.; Chandu, K.; et al. 2024. RewardBench: Evaluating Reward Models for Language Modeling. *arXiv preprint arXiv: 2403.13787*.

Li, J.; Sun, S.; Yuan, W.; Fan, R.-Z.; hai zhao; and Liu, P. 2024a. Generative Judge for Evaluating Alignment. In *Proc. of ICLR*.

Li, T.; Chiang, W.-L.; Frick, E.; Dunlap, L.; Wu, T.; Zhu, B.; Gonzalez, J. E.; and Stoica, I. 2024b. From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and Benchbuilder Pipeline. *arXiv preprint arXiv: 2406.11939*.

Li, X.; Yu, P.; Zhou, C.; Schick, T.; Levy, O.; Zettlemoyer, L.; Weston, J. E.; and Lewis, M. 2024c. Self-Alignment with Instruction Backtranslation. In *Proc. of ICLR*.

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. AlpacaEval: An Automatic Evaluator of Instruction-following Models. https://github.com/tatsu-lab/alpaca_eval.

Lin, B. Y.; Deng, Y.; Chandu, K.; Brahman, F.; Ravichander, A.; Pyatkin, V.; Dziri, N.; Bras, R. L.; and Choi, Y. 2024. WildBench: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild. *arXiv preprint arXiv: 2406.04770*.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; and Roberts, A. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proc. of ICML*, volume 202 of *Proceedings of Machine Learning Research*, 22631–22648.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Proc. of NeurIPS*.

Mehri, S.; and Eskenazi, M. 2020. Unsupervised Evaluation of Interactive Dialog with DialoGPT. In Pietquin, O.; Muresan, S.; Chen, V.; Kennington, C.; Vandyke, D.; Dethlefs, N.; Inoue, K.; Ekstedt, E.; and Ultes, S., eds., *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 225–235.

OpenAI. 2023. GPT-4 Technical Report. *PREPRINT*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Proc. of NeurIPS*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Proc. of NeurIPS*.

Rasul, K.; Beeching, E.; Tunstall, L.; von Werra, L.; and Sanseviero, O. 2024. Preference Tuning LLMs with Direct Preference Optimization Methods.

Tao, Z.; Lin, T.-E.; Chen, X.; Li, H.; Wu, Y.; Li, Y.; Jin, Z.; Huang, F.; Tao, D.; and Zhou, J. 2024. A Survey on Self-Evolution of Large Language Models. *arXiv preprint arXiv: 2404.14387*.

Tunstall, L.; Beeching, E.; Lambert, N.; Rajani, N.; Rasul, K.; Belkada, Y.; et al. 2023. Zephyr: Direct Distillation of LM Alignment. *arXiv preprint arXiv: 2310.16944*.

Wang, H.; Xiong, W.; Xie, T.; Zhao, H.; and Zhang, T. 2024a. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. *arXiv preprint arXiv: 2406.12845*.

Wang, Y.; Ivison, H.; Dasigi, P.; Hessel, J.; Khot, T.; Chandu, K.; Wadden, D.; MacMillan, K.; Smith, N. A.; Beltagy, I.; and Hajishirzi, H. 2023. How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Proc. of NeurIPS*.

Wang, Z.; Dong, Y.; Zeng, J.; Adams, V.; Sreedhar, M. N.; Egert, D.; Delalleau, O.; Scowcroft, J.; Kant, N.; Swope, A.; and Kuchaiev, O. 2024b. HelpSteer: Multi-attribute Helpfulness Dataset for SteerLM. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proc. of NAACL-HLT*, 3371–3384.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022. Finetuned Language Models are Zero-Shot Learners. In *Proc. of ICLR*.

Wu, T.; Yuan, W.; Golovneva, O.; Xu, J.; Tian, Y.; Jiao, J.; Weston, J.; and Sukhbaatar, S. 2024a. Meta-Rewarding Language Models: Self-Improving Alignment with LLM-as-a-Meta-Judge. *arXiv preprint arXiv: 2407.19594*.

Wu, Y.; Sun, Z.; Yuan, H.; Ji, K.; Yang, Y.; and Gu, Q. 2024b. Self-Play Preference Optimization for Language Model Alignment. *arXiv preprint arXiv: 2405.00675*.

Wu, Z.; Hu, Y.; Shi, W.; Dziri, N.; Suhr, A.; Ammanabrolu, P.; Smith, N. A.; Ostendorf, M.; and Hajishirzi, H. 2023. Fine-Grained Human Feedback Gives Better Rewards for Language Model Training. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Proc. of NeurIPS*.

Xu, C.; Guo, D.; Duan, N.; and McAuley, J. 2023a. Baize: An Open-Source Chat Model with Parameter-Efficient Tuning on Self-Chat Data. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proc. of EMNLP*, 6268–6278.

Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; and Jiang, D. 2023b. WizardLM: Empowering Large Language Models to Follow Complex Instructions. *arXiv preprint arXiv: 2304.12244*.

Xu, J.; Lee, A.; Sukhbaatar, S.; and Weston, J. 2023c. Some things are more CRINGE than others: Preference Optimization with the Pairwise Cringe Loss. *arXiv preprint arXiv: 2312.16682*.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; et al. 2024. Qwen2 Technical Report. *arXiv preprint arXiv: 2407.10671*.

Ye, S.; Kim, D.; Kim, S.; Hwang, H.; Kim, S.; Jo, Y.; Thorne, J.; Kim, J.; and Seo, M. 2024. FLASK: Fine-grained Language Model Evaluation based on Alignment Skill Sets. In *Proc. of ICLR*.

Yuan, W.; Pang, R. Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-Rewarding Language Models. *arXiv preprint arXiv: 2401.10020*.

Zhang, C.; D'Haro, L. F.; Chen, Y.; Zhang, M.; and Li, H. 2024a. A Comprehensive Analysis of the Effectiveness of Large Language Models as Automatic Dialogue Evaluators. *Proc. of AAAI*.

Zhang, C.; Tang, C.; Chong, D.; Shi, K.; Tang, G.; Jiang, F.; and Li, H. 2024b. TS-Align: A Teacher-Student Collaborative Framework for Scalable Iterative Finetuning of Large Language Models. In *Findings of EMNLP 2024*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; et al. 2023a. A Survey of Large Language Models. *arXiv preprint arXiv: 2303.18223*.

Zhao, Y.; Joshi, R.; Liu, T.; Khalman, M.; Saleh, M.; and Liu, P. J. 2023b. SLiC-HF: Sequence Likelihood Calibration with Human Feedback. *arXiv preprint arXiv: 2305.10425*.

Zheng, L.; Chiang, W.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023a. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Proc. of NeurIPS*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Li, T.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Li, Z.; Lin, Z.; Xing, E. P.; Gonzalez, J. E.; Stoica, I.; and Zhang, H. 2023b. LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset. *arXiv preprint arXiv: 2309.11998*.

Zheng, Y.; Zhang, R.; Zhang, J.; Ye, Y.; Luo, Z.; Feng, Z.; and Ma, Y. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. *arXiv preprint arXiv: 2403.13372*.

Zhu, B.; Frick, E.; Wu, T.; Zhu, H.; and Jiao, J. 2023. Starling-7B: Improving LLM Helpfulness and Harmlessness with RLAIF.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine-Tuning Language Models from Human Preferences. *arXiv preprint arXiv: Arxiv-1909.08593*.