ULTRAFEEDBACK: Boosting Language Models with Scaled AI Feedback

Ganqu Cui [*1] **Lifan Yuan** [*1 2] **Ning Ding** [1] **Guanming Yao** [1 3] **Bingxiang He** [1] **Wei Zhu** [4] **Yuan Ni** [4] **Guotong Xie** [4] **Ruobing Xie** [5] **Yankai Lin** [6] **Zhiyuan Liu** [1] **Maosong Sun** [1 7]

Abstract

Learning from human feedback has become a pivot technique in aligning large language models (LLMs) with human preferences. However, acquiring vast and premium human feedback is bottlenecked by time, labor, and human capability, resulting in small sizes or limited topics of current datasets. This further hinders feedback learning as well as alignment research within the open-source community. To address this issue, we explore how to go beyond human feedback and collect high-quality *AI feedback* automatically for a scalable alternative. Specifically, we identify **scale and diversity** as the key factors for feedback data to take effect. Accordingly, we first broaden instructions and responses in both amount and breadth to encompass a wider range of user-assistant interactions. Then, we meticulously apply a series of techniques to mitigate annotation biases for more reliable AI feedback. We finally present ULTRAFEEDBACK, a large-scale, high-quality, and diversified AI feedback dataset, which contains over 1 million GPT-4 feedback for 250k user-assistant conversations from various aspects. Built upon ULTRAFEEDBACK, we align a LLaMA-based model by best-of-n sampling and reinforcement learning, demonstrating its exceptional performance on chat benchmarks. Our work validates the effectiveness of scaled AI feedback data in constructing strong open-source chat language models, serving as a solid foundation for future feedback learning research.

[*]Equal contribution [1]NLP Group, DCST, IAI, BNRIST, Tsinghua University [2]University of Illinois Urbana-Champaign [3]ModelBest.Inc [4]PingAn Technology [5]Tencent [6]Renmin University of China [7]Jiangsu Collaborative Innovation Center for Language Ability. Correspondence to: Ganqu Cui <cgq22@mails.tsinghua.edu.cn>, Lifan Yuan <lifan4@illinois.edu>, Wei Zhu <zhuwei972@pingan.com.cn>, Zhiyuan Liu <liuzy@tsinghua.edu.cn>, Maosong Sun <sms@tsinghua.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Large language models (LLMs) (OpenAI, 2022; 2023) have demonstrated proficiency in generating fluent text as well as solving various language-oriented tasks. Trained on massive corpora through likelihood maximization techniques, these LLMs have equipped the ability to execute diverse tasks in response to user directives (Ouyang et al., 2022; Wei et al., 2022a; Sanh et al., 2022). Unfortunately, relying solely on imitation learning during training leads to well-known issues - LLMs may generate convincing but incorrect or unsafe content that deviates from human preferences (Stiennon et al., 2020; Perez et al., 2022). To further align LLMs with human preferences, learning from human feedback (Ouyang et al., 2022; Askell et al., 2021; Bai et al., 2022a; Touvron et al., 2023b) has been introduced and widely adopted by leading corporations. Over a period, feedback learning has been widely applied to closed-source models but scarcely used in open-source models.

Many factors hinder the implementation of feedback learning in the open-source community, but the first and primary issue is *data*. Preference data, which rates and compares different responses given the same prompt, is central to feedback learning. When scaled sufficiently, preference data reflects the intrinsic values of the annotators. Such annotators are often assumed, by default, to be human beings who can provide the most flexible and accurate supervision signals, yet the data they generate is severely bounded by factors like financial resources, time, and knowledge. As a result, existing preference datasets are either small in scale (Wu et al., 2023) or limited on specific tasks (Stiennon et al., 2020; Nakano et al., 2021). To this end, more efficient and principled methods to scale preference data are on the horizon.

This study aims to scale feedback data in an efficient manner. Specifically, we explore *AI feedback* (Bai et al., 2022b; Lee et al., 2023), which substitutes human annotators with advanced LLMs. Compared with human feedback, AI feedback is more *scalable*, which means (1) it is easier to collect and expand with lower cost; (2) its quality improves as the LLM annotators become more capable. In previous research, it is shown that advanced AI systems are capable of conducting chatbot evaluations (Dubois et al., 2023; Zheng et al., 2023a), giving textual critiques (Ye et al., 2023; Wang

et al., 2023c), or assisting human annotators (Saunders et al., 2022). However, open-source LLMs have not yet benefited from AI feedback through the lens of feedback learning.

This paper establishes a comprehensive AI feedback collection pipeline. Besides scalability, we prioritize **diversity** of both instructions and responses for holistic language model alignment. Particularly, we compile a diverse array of over $60,000$ instructions and 17 models from multiple sources to produce comparative conversations in broad topics and quality. Then, we adopt a bunch of techniques to alleviate annotation biases and improve feedback quality to the greatest extent. These include (1) decomposing annotation documents into four different aspects, namely instruction-following, truthfulness, honesty, and helpfulness, to reduce ambiguity; (2) providing objective grading criteria and reference responses for score calibration; (3) asking GPT-4 for detailed textual critique before scores as chain-of-thought (Wei et al., 2022b) rationales. Comprehending all above, we finally build ULTRAFEEDBACK, a million-scale AI feedback dataset for aligning open-source LLMs.

We comprehensively validate the advantage of AI feedback in boosting open-source models with ULTRAFEEDBACK. By fine-tuning a LLaMA2-13B model (Touvron et al., 2023b), we build a state-of-the-art reward model UltraRM, which significantly outperforms existing open-source reward models. Based on UltraRM, we enhance a powerful open-source model UltraLM (Ding et al., 2023; Touvron et al., 2023a) with best-of-n sampling and PPO. Experiments show that both strategies improve the model dramatically. Moreover, we fine-tune a critique model that could criticize and judge model responses. We also conduct a detailed analysis of the consistency and inconsistency between AI and human feedback.

To summarize, our contributions are three-fold: (1) To the best of our knowledge, we for the first time demonstrate the beneficial effect of scaled AI feedback on open-source chat LLMs. (2) We establish a systematic and sizable pipeline to collect high-quality and diversified AI feedback. (3) We release a suite of resources for feedback learning research, including a dataset, reward model, and critique model.

2. ULTRAFEEDBACK

2.1. Overview

Inspired by the data engineering principles in supervised fine-tuning (Ding et al., 2023; Chiang et al., 2023; Xu et al., 2023), we identify scalability and diversity as pivot factors of the overall generalizability of preference data. We argue that existing preference data suffer from satisfying either one of the two factors. To be specific, human feedback collection usually relies on human annotators to compare a pair of completions (Stiennon et al., 2020; Nakano et al., 2021; Ouyang et al., 2022; Bai et al., 2022a). Thus, the data is hard to scale up due to time and budget constraints, especially for open-source researchers. On the other hand, existing AI feedback approaches (Bai et al., 2022b; Lee et al., 2023) reduce human involvement and enjoy scalability via capable LLMs, but they are limited to specific domains (Bai et al., 2022b; Lee et al., 2023) or forms (Ye et al., 2023) and hence lack the necessary diversity to boost LM performance under broader contexts.

To this end, we take into account scalability and diversity in all three stages of the preference data collection process: collecting instructions, sampling completions, and annotating comparison pairs. The overview of the data collection pipeline is shown in Figure 1. Firstly, we collect a large-scale and diversified instruction set to enhance LLMs' capabilities from four aspects: (1) *Follow Instructions*: LLMs should respond to humans without deviating from the requirements. (2) *Helpful and Informative*: LLMs should provide useful and correct answers to address the given problems. (3) *Truthful*: LLMs' output should be grounded in the instructions and real-world knowledge, and avoid introducing any self-contradiction. (4) *Honesty*: LLMs should know what they (don't) know and express uncertainty towards the given problem. For the second stage, to avoid the sameness of comparison responses, we build a pool of distinct models at different capability levels to sample completions. Finally, to overcome the issues concerning scalability (Nakano et al., 2021; Stiennon et al., 2020) and quality (Ethayarajh et al., 2022), we seek scalable AI feedback from GPT-4, and explore several techniques to improve the reliability. Next, we will introduce our data construction pipeline in detail.

2.2. Instruction Collection

We select instructions that target four distinct but all-important abilities of language models, namely instruction-following, truthfulness, honesty, and helpfulness. Specifically, we include all instructions from TruthfulQA (Lin et al., 2022) and FalseQA (Hu et al., 2023) training set for truthfulness. For instruction-following and helpfulness, we randomly sample 10k instructions from Evol-Instruct (Xu et al., 2023) and UltraChat (Ding et al., 2023) respectively, and sample 20k from ShareGPT (Chiang et al., 2023). We finally include FLAN (Longpre et al., 2023) to improve LLMs' helpfulness in various NLP tasks due to the task diversity within FLAN. We adopt a stratified sampling strategy following (Mukherjee et al., 2023), randomly picking 3k instructions from the "CoT" subset and sampling 10 instructions per task for the other three subsets, while excluding those with overly long instructions. In particular, honesty will be assessed by TruthfulQA and FLAN as they both contain reference answers, based on which it is easier for the annotator to judge if the uncertainty expressed in LLMs' responses calibrates with the accuracy. We then conduct

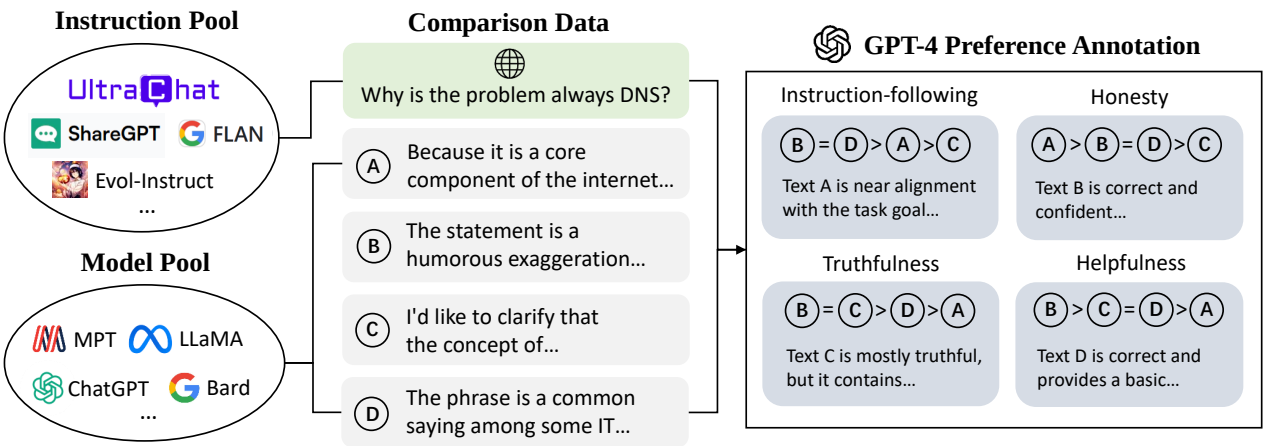

Figure 1. ULTRAFEEDBACK construction process. We sample instructions and models from large pools to guarantee diversity, then query GPT-4 with detailed illustrations for fine-grained and high-quality annotations in both textual and numerical formats.

a data contamination detection (see Appendix B). Finally, we obtain $63,967$ instructions of various types from the six publicly available high-quality datasets.

2.3. Completion Sampling

To guarantee that the collected responses are dissimilar and well-distributed, we include different models to generate completions for each instruction. To alleviate the potential spurious correlation between text styles and response quality within the dataset, we introduce intervention by selecting not only different series of models at different levels, but also models with different model sizes, architectures, and training data within the same model series. This strategy enables one type of text style to present responses of different quality levels, namely the response of one series of models may be better or worse than another depending on model sizes, thus avoiding the establishment of spurious correlations.

Specifically, we set up a pool of 17 models: (1) For commercial models, we choose GPT-4, `gpt-3.5-turbo` (ChatGPT), and Bard [1]; (2) For LLaMA-series, we choose UltraLM-13B/65B (Ding et al., 2023), WizardLM-7B-v1.1/13B-v1.2/70B-v1.1 (Xu et al., 2023), Vicuna-33B-v1.3 (Chiang et al., 2023), LLaMA2-7B/13B/70B-Chat (Touvron et al., 2023b), and Alpaca-7B (Taori et al., 2023); (3) For Non-LLaMA series, we choose MPT-30B-Chat (MosaicML, 2023), Falcon-40B-Instruct (Almazrouei et al., 2023), StarChat (Tunstall et al., 2023), and Pythia-12B (Biderman et al., 2023). We randomly sample four different models from the pool to complete each instruction.

To further improve diversity in model responses, we elicit distinct model behaviors by adding different principles before completing each instruction. Following Sun et al. (2023) and Mukherjee et al. (2023), we first hand-craft one principle for each aspect and then automize the procedure by invoking GPT-4 to curate another ten based on the human-written example. According to dataset characteristics, each data source is assigned with different principle prompts. We randomly sample a corresponding principle for each completion and add it to the system prompt to induce model behaviors. The principles can be found in Appendix G.1, and the effects of different principles are plotted in Figure 6.

2.4. AI Feedback Annotation

After generating $255,864$ model completions based on the $63,967$ instructions, we employ GPT-4 to provide two types of feedback for each completion: (1) scalar scores that indicate the fine-grained quality regarding multiple aspects, and (2) textual critique that gives detailed guidance on how to improve the completion. These lead to over 1 million feedback data in total.

Preference Annotation. Regarding the potential subjectivity and randomness of GPT-4 annotation, we apply four techniques to improve the annotation quality: (1) **Decomposition**. To reduce ambiguity and the difficulty of annotation, we decompose the overall quality assessment into four fine-grained assessments, namely instruction-following, truthfulness, honesty, and helpfulness. (2) **Standard**. For each aspect, we provide GPT-4 with detailed documentation of scores from 1 to 5 for reference, thus avoiding variable and subjective standards. See Appendix G.2 for an example. (3) **Reference**. To prevent inconsistency ratings across different runs, we wrap one instruction and all its completions into the prompt and ask GPT-4 to score four completions simultaneously to reduce randomness. (4) **Rationale**. Besides scoring each response, GPT-4 is required to generate a rationale on how the response should be scored according to the documentation. Combining all the techniques, we finally have four fine-grained scalar scores and rationales for each response.

[1] https://bard.google.com/

Table 1. Statistics of existing preference and critique datasets. The average length refers to the number of tokens.

Dataset	# Convs	Prompt Length	Response Length	Critique Length	Fine-Grained?	Feedback Format	# Pairs	# Critique	Annotator
Preference Dataset									
OASST1	35,905	167.6	221.1	-	✗	Scalar	17,966	-	Human
OpenAI WebGPT	38,925	50.9	188.2	-	✗	Scalar	19,578	-	Human
Anthropic Helpful	118,263	185.7	94.6	-	✗	Ranking	118,263	-	Human
OpenAI Summ.	60,674	**326.4**	36.6	-	✓	Scalar	92,858	-	Human
QA Feedback	11,378	155.8	107.9	-	✓	Scalar	17,118	-	Human
Critique Dataset									
SelFee	178,331	100.3	243.9	89.4	✓	Text	-	**316,026**	AI
Shepherd	1,316	95.3	97.6	67.2	✓	Text	-	1,317	Human
ULTRAFEEDBACK	**255,864**	185.1	**305.3**	**143.1**	✓	Scalar & Text	**340,025**	255,864	AI

Critique Generation. Besides scalar reward, we also seek textual critique from GPT-4. We prompt GPT-4 to act as a tutor and provide detailed suggestions specified for each completion to help models improve rather than propose answers directly. Different from the above comparison-oriented annotations, critique prompts are generated separately from an overall perspective for each completion. The prompts can be found in Appendix G.2.

2.5. Dataset Statistics

We compare ULTRAFEEDBACK with current open-source datasets in Table 1. ULTRAFEEDBACK stands out to be the largest one among all preference and critique datasets, which is at least twice as large as other datasets. Also, its completions and critiques are the longest. Moreover, we highlight that ULTRAFEEDBACK is **the only dataset that provides both scalar preferences and textual feedback**, enabling it to serve as a preference and critique dataset simultaneously. Overall, ULTRAFEEDBACK outperforms previous datasets in both scale and diversity, and we also validate its high quality by experiment in Section 3.

2.6. ULTRAFEEDBACK-Powered Models

Based on ULTRAFEEDBACK, we develop UltraRM, an advanced open-source reward model that provides preferences for AI responses given user instructions. Additionally, we train a critique model UltraCM from the textual feedback in ULTRAFEEDBACK. UltraCM could interact with human and AI assistants more flexibly in text.

UltraRM. For reward modeling, we train UltraRM based on LLaMA2-13B (Touvron et al., 2023b). Specifically, we train three versions of UltraRM. We mix several open-source datasets with ULTRAFEEDBACK to train UltraRM. The open-source datasets include Stanford SHP (Ethayarajh et al., 2022), OpenAI Summarization (Stiennon et al., 2020), and Anthropic Helpful (Bai et al., 2022a). To validate the quality of UltraFeedback, we also train one model with *merely* the fine-grained scores of this dataset, i.e. averaging the preference scores in each aspect to get a final reward score. Further, to compare the effectiveness of the fine-grained scores and overall scores, we replace the fine-grained scores in UltraRM with the assessment ratings in critique generation, while remaining the open-source datasets. The details for dataset processing can be found in Appendix E.1. We keep the training strategy, including loss objective and training hyperparameters, exactly the same as Touvron et al. (2023b).

UltraCM. We also train a critique model stemming from ULTRAFEEDBACK to boost future research in learning from textual feedback (Wang et al., 2023d). UltraCM has the same initialization as UltraRM but is trained solely on ULTRAFEEDBACK critique data, i.e. $255,864$ textual feedback in total. Given a response, we fine-tune the model to give a corresponding critique that judges the response, figures out flaws, and provides suggestions for improvement.

3. Experiments

To validate the effect of AI feedback, we first evaluate UltraRM on human preference benchmarks in Section 3.1. Next, we test UltraRM in enhancing chat language models with two strategies, namely best-of-n sampling (Section 3.2) and reinforcement learning (Section 3.3). Finally, we evaluate the feedback quality of UltraCM in Appendix E.3.

3.1. Reward Modeling

Setup. To evaluate how UltraRM aligns with human preference, we conduct experiments on four **human annotated** preference datasets, OpenAI WebGPT (Nakano et al., 2021), OpenAI Summarization (Stiennon et al., 2020), Anthropic HH-RLHF (Bai et al., 2022a), and Standford SHP. On each dataset, we calculate the rewards of two responses for one prompt and predict which one is more preferred. We compare our UltraRM-UF, UltraRM-Overall, and Ul-

Table 2. Reward modeling accuracy (%) results. We compare our UltraRM with baseline open-source reward models. LLaMA2 results are taken from (Touvron et al., 2023b). The highest results are in **bold** and the second highest scores are underlined.

Model	Backbone Model	Open?	Anthropic Helpful	OpenAI WebGPT	OpenAI Summ.	Stanford SHP	Avg.
Moss	LLaMA-7B	✓	61.3	58.1	59.0	54.6	58.3
Ziya	LLaMA-7B	✓	61.4	61.8	60.3	57.0	60.1
OASST	DeBERTa-v3-large	✓	67.6	-	71.8	53.9	-
SteamSHP	FLAN-T5-XL	✓	55.4	62.6	48.4	51.6	54.5
LLaMA2 Helpfulness	LLaMA2-70B	✗	**72.0**	-	**75.5**	**80.0**	-
UltraRM	LLaMA2-13B	✓	71.0	**65.2**	74.0	73.7	**71.0**
w/ Only ULTRAFEEDBACK	LLaMA2-13B	✓	66.7	65.1	66.8	68.4	66.8
w/ Overall Score	LLaMA2-13B	✓	71.0	62.0	73.0	73.6	69.9

traRM with open-source baselines, including Moss (Zheng et al., 2023b), Ziya (IDEA-CCNL, 2021), OASST [2], and SteamSHP (Ethayarajh et al., 2022). We also report the results in LLaMA2 (Touvron et al., 2023b), although their reward models are not released.

Results. The preference prediction accuracy results are reported in Table 2. As we can see, the UltraRM series outperforms baseline reward models except for the closed LLaMA2 reward model (much larger) by a large margin, indicating that UltraRM series are the **best open-source reward models**. Notably, our reward model can still surpass all other baselines even without mixing open-source datasets. These results reveal that, ULTRAFEEDBACK is highly consistent with human preference, and its high quality as well as diversity enable strong out-of-distribution generalization. On average, the model trained with only ULTRAFEEDBACK outperforms open-source baseline models by over 6.3 percent in accuracy, while mixing open-source datasets with overall scores and fine-grained scores of ULTRAFEEDBACK achieves 3.1 and 4.2 percent more improvement respectively.

We highlight that the OpenAI WebGPT dataset has no training and test splits, and neither most baselines nor we train reward models on this dataset[3], making it a fair benchmark to evaluate the generalization ability of reward models. Obviously, UltraRM series are significantly better, reaching 2.6% absolute points improvement over baselines. Another intriguing finding is that adding open-source datasets has a minor effect on the WebGPT dataset, which again proves the transferability advantage of ULTRAFEEDBACK. On another benchmark Stanford SHP, UltraRM also achieves remarkable performance.

A noteworthy finding is that, despite exhibiting comparably on the other three datasets, the reward model trained with overall scores discernably lags behind the other two variants on WebGPT. There can be two potential explanations for this observation. First, fine-grained annotation, which scores model outputs from different aspects respectively, provides a more precise assessment for each completion than aggregating evaluation into an overall number. Second, in the overall quality annotation process, each sample is sent to GPT-4 separately whereas, in fine-grained rating, all four completions are scored at the same time, which may provide GPT-4 with cross-references and prevent it from applying inconsistent standards, reducing the impact of randomness. These superiorities demonstrate the high quality of our fine-grained preference data, and we advocate future work to adopt the fine-grained annotation schema and rate multiple completions at one time.

[2] https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2

[3] The OASST and LLaMA2 Helpfulness reward model used WebGPT dataset for training. To prevent data leakage, we do not report their performance on WebGPT.

3.2. Best-of-n Experiments

Figure 2. Win rate against `text-davinci-003` on AlpacaEval benchmark. We sample n responses and choose the one with the highest reward.

Setup. To verify that UltraRM could serve as a good indicator of response quality, we conduct best-of-n experiments. On the AlpacaEval benchmark, we randomly sample 16 completions from the original UltraLM-13B and calculate their corresponding rewards. We then select the best-of-

Table 3. Head-to-head comparison results on three public benchmarks. The baseline is `text-davinci-003` in AlpacaEval and `gpt-3.5-turbo` in Evol-Instruct and UltraChat. The judge is GPT-4. The highest win rates are in **bold**.

Model	Size	AlpacaEval Win (%)	Evol-Instruct Win / Tie / Lose (%)	UltraChat Win / Tie / Lose (%)	Average Win (%)
ChatGPT	-	89.4	-	-	-
			LLaMA2		
Vicuna-13B-v1.5	13B	-	33.0 / 23.9 / 43.1	34.5 / 38.2 / 27.3	-
LLaMA2-13B-Chat	13B	81.1	44.1 / 11.9 / 44.0	53.5 / 21.3 / 25.2	59.5
WizardLM-13B-v1.2	13B	89.2	55.5 / 17.4 / 27.1	59.7 / 25.5 / 14.8	68.1
OpenChat-13B-v3.2super	13B	89.5	55.5 / 11.0 / 33.5	58.7 / 26.7 / 14.5	67.9
LLaMA2-70B-Chat	70B	**92.7**	56.4 / 13.8 / 29.8	54.0 / 28.6 / 17.4	67.7
			LLaMA		
UltraLM-13B	13B	80.7	39.9 / 14.7 / 45.4	38.2 / 34.8 / 27.0	52.9
Vicuna-13B-v1.3	13B	82.1	36.7 / 17.4 / 45.9	41.3 / 33.2 / 25.5	53.4
WizardLM-13B-v1.1	13B	86.3	54.1 / 14.7 / 31.2	56.1 / 26.0 / 17.9	65.5
Vicuna-33B-v1.3	33B	89.0	50.0 / 17.0 / 33.0	57.7 / 25.7 / 16.6	65.6
UltraLM-13B-PPO	13B	86.3	**57.8** / 10.1 / 32.1	**64.9** / 15.6 / 19.5	**69.7**

$\{1, 2, 4, 8, 16\}$ completions as the final response. The sampling parameters are set to temperature $= 1$ and top-$p = 1$.

Results. We present results in Figure 2. Apparently, we can see the win rate on AlpacaEval increases proportionally with rewards. This validates that our UltraRM gives rigorous rewards that reflect the overall response quality. Notably, the best-of-n sampling strategy is surprisingly effective. The initial UltraLM-13B model achieves a 76.53% win rate for a single sampling, and a simple best-of-2 sample increases the win rate to 84.64%. With more samples, we can get even more high-quality responses, and the final best-of-16 win rate hits 91.54%. The best-of-n sampling is universally applicable across models and tasks, which enhances models without training. Please refer to Appendix F.2 for cases.

3.3. PPO Experiments

Setup. Given the state-of-the-art UltraRM, we aim to push the upper bound of open-source chat language models with RLAIF. Specifically, we perform PPO over UltraLM-13B (Ding et al., 2023) to get its PPO version, UltraLM-13B-PPO. We tune UltraLM for 80 iterations on the ULTRAFEEDBACK prompts. In each iteration, we collect 512 samples and update the policy model with a mini-batch size of 64. The learning rate is fixed at 1e-6.

Baselines. We compare UltraLM-13B-PPO with leading open-source models and proprietary models, including LLaMA2-Chat (Touvron et al., 2023b), Vicuna (Chiang et al., 2023), WizardLM (Xu et al., 2023), OpenChat (Wang et al., 2023a), and ChatGPT (OpenAI, 2022).

Benchmarks. We conduct experiments on three public benchmarks, namely AlpacaEval (Li et al., 2023), Evol-Instruct (Xu et al., 2023), and UltraChat (Ding et al., 2023). On each benchmark, we ask GPT-4 to judge which response is better given the same instruction. AlpacaEval adopts `text-davinci-003` as the competitor model, while we compete with `gpt-3.5-turbo` on Evol-Instruct and UltraChat. To avoid position bias, we randomly switch the comparing responses. For all models, we use the same decoding parameter with temperature $= 0.7$ and top-$p = 1$.

Results. We report experiment results in Table 3. We take the official results on the AlpacaEval leaderboard for baseline models and conduct evaluations by ourselves for other results. Overall, our UltraLM-13B-PPO achieves the highest average win rate on the three benchmarks, outperforming all other open-source models. Among LLaMA-based models, UltraLM-13B-PPO overtakes other models by at least 3.6 percent on average. Even when compared with the much larger LLaMA2-70B-Chat model, our model still holds the advantage, illustrating the huge benefit of RLAIF alignment. Our model also reaches the highest win rate on two of the benchmarks, Evol-Instruct and UltraChat, against the more powerful `gpt-3.5-turbo`. It is worth noting that, compared with the original UltraLM-13B, the PPO process benefits the model greatly, leading to a 16.8 percent enhancement. We provide cases in Appendix F.3.

4. Agreement with Human Preferences

The inclusivity of human preferences is known to be hard to capture (Dubois et al., 2023). Heavily relying on AI feedback, it is essential to measure and monitor the agreement between AI and human preferences. In this section, we conduct experiments to see (1) to what extent AI annotations are consistent with human preferences (Section 4.1) and (2) how reliable AI evaluations are (Section 4.2).

4.1. Annotation Consistency

In Section 3.1, we show that the reward models trained on ULTRAFEEDBACK could predict human preference accurately. To further analyze to what extent AI feedback could capture human preference, we randomly sample 400 comparison pair from ULTRAFEEDBACK, AlpacaEval, Evol-Instruct and UltraChat test sets (100 each) and ask 3 independent annotators to compare those pairs (win/tie/lose). The annotators are undergraduate and graduate students. We present the agreement ratio between GPT-4 and annotators, as well as annotators themselves in Table 4. On average, GPT-4 judge exhibits 59.7% agreement rate with human labelers, which matches previous human evaluation on MT-Bench (Zheng et al., 2023a). We also observe similar agreement rates among annotators. Notably, the agreement between GPT-4 and the majority votes of three annotators raises to 68.6%, meaning that GPT-4 better reflects the collective human preferences.

Table 4. Agreement between judges on 400 samples from ULTRAFEEDBACK, AlpacaEval, Evol-Instruct, and UltraChat test sets . A-1, A-2, A-3 are three human judges. "Majority" stands for the agreement between each judge and other three judges's majority votes. We include tie votes and the random agreement is 33%.

Judge	A-1	A-2	A-3	Average	Majority
GPT-4	59.2%	60.8%	59.1%	**59.7%**	**68.6%**
A-1	-	58.1%	54.7%	57.3%	60.3%
A-2	58.1%	-	55.4%	58.1%	63.3%
A-3	54.7%	55.4%	-	56.4%	62.0%

4.2. Reliability of AI Evaluation

We first supplement another AI evaluation using Claude-3 Sonnet (Anthropic, 2024) to investigate the agreement among different series of AI models. The prompts are the same as the GPT-4 evaluation. Then, we compare both of our AI evaluation results with human annotations to examine if AI evaluations reliably correlate with humans. Particularly, we use the majority votes of the three annotators and filter out samples with all different votes.

We present GPT-4, Claude-3, and human evaluation results on the remaining 266 pairs in Table 5. Overall, Claude-3 shares the same trend as GPT-4 and further increases our models' win rates. Human evaluations are mostly consistent with GPT-4, giving a 64.3% against 67.3% average winning rate. We notice that human labelers tend to assign more ties than GPT-4, leading to slightly lower winning rates.

For fine-grained analysis, we categorize the evaluated samples into reasoning, writing, and QA tasks. The categorical comparison results are presented in Figure 3. It is shown that human evaluations are mostly consistent with GPT-4, where they both prefer our models on writing and QA tasks. On reasoning tasks including coding, math, and logic, human and GPT-4 judgments diverge on ties and losses, where GPT-4 gives fewer ties but more losses. To delve into the discrepancy deeper, we ask another expert labeler to determine the ground truth answer for each question. In this sense, a model wins when it gives the correct answer while the other does not, and vice versa. The two models tie when they both successfully or unsuccessfully answer the question. The final win/tie/lose rate comes at $42.1\%/26.3\%/31.6\%$ and closely matches human evaluations. The GPT-4 judge, in this case, potentially underestimated our model's reasoning performance and still has space to improve.

Table 5. Human evaluation results. We use majority votes from three human judges and compare GPT-4, Claude-3, and human evaluations on the same 266 samples.

Judge	AlpacaEval Win (%)	Evol-Instruct Win / Tie / Lose (%)	UltraChat Win / Tie / Lose (%)	Avg. Win (%)
GPT-4	83.9	57.1 / 8.8 / 34.1	61.0 / 17.1 / 21.9	67.3
Claude-3	95.1	59.6 / 1.4 / 39.0	73.5 / 6.8 / 19.7	76.1
Human	78.5	68.1 / 17.6 / 14.3	46.3 / 19.5 / 34.1	64.3

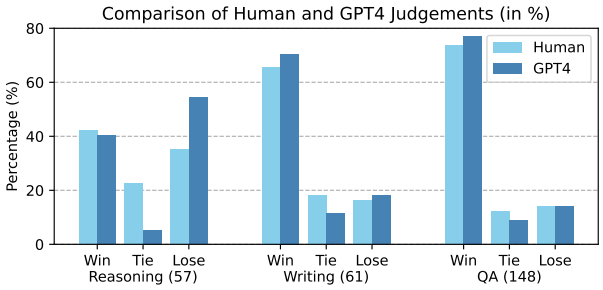

Figure 3. Catrgorical comparison of human and GPT-4 judgments. Human judgments are majority votes from three annotators. Sample numbers of each category are in parentness.

5. Analysis

In this section, we further analyze how ULTRAFEEDBACK enhances language models on different subjects (Section 5.1) and tasks (Section 5.2).

5.1. Question Type Breakdown

Figure 4 reports the UltraLM-13B-PPO and UltraLM-13B scores on different question types versus `gpt-3.5-turbo` on the Evol-Instruct test set. We observe that UltraLM-13B-PPO overtakes ChatGPT on 22/29 subjects, especially on writing-related tasks such as academic writing. Our model is also well-aligned with human values, getting higher scores on toxicity, ethics, and TruthfulQA. On some difficult subjects like roleplay, reasoning, and counterfactual, our model is still on par with ChatGPT, indicating the strong advanced model capability. Compared with the original UltraLM-13B, PPO boosts the model in multiple aspects, including professional knowledge (economy, chemistry, music, literature) and reasoning ability (reasoning, complex format, code generation, math). Meanwhile, our model falls

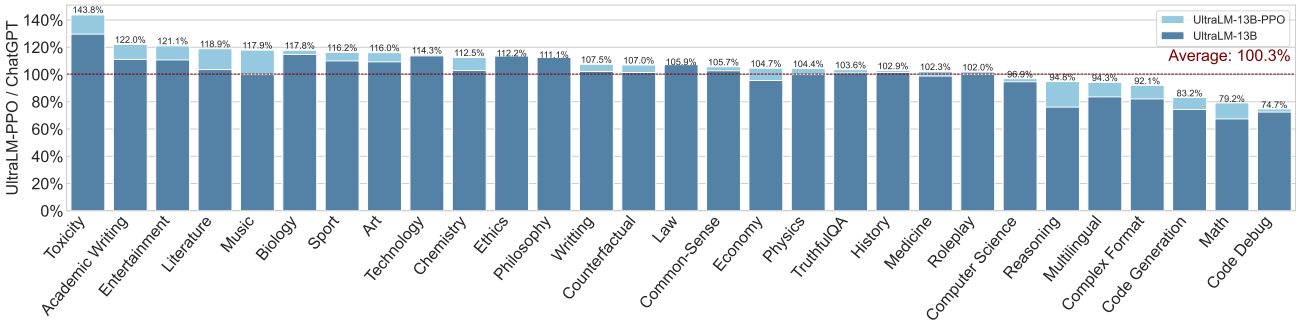

Figure 4. Comparison results between UltraLM-13B-PPO, UltraLM-13B, and `gpt-3.5-turbo` on Evol-Instruct test set, where `gpt-3.5-turbo` scores are 100%.

Table 6. Exact match scores (%) for UltraLM-13B and UltraLM-13B-PPO on model capability benchmarks.

Model	BoolQ	HellaSwag	RACE-h	RACE-m	MultiRC	TriviaQA	NQ	PIQA	OBQA	ARC-E	ARC-C	Avg.
UltraLM-13B	85.0	59.8	66.1	73.5	83.2	50.8	19.4	73.5	57.0	76.1	51.5	63.3
UltraLM-13B-PPO	83.5	62.6	66.8	74.2	83.7	52.5	22.1	74.9	57.0	76.1	53.9	64.3

behind `gpt-3.5-turbo` on math and code-related tasks, which might be attributed to the limitation of base model ability and the lack of relevant data in ULTRAFEEDBACK. Table 9 in Appendix E.5 provides additional results on the UltraChat test set and reaches the same conclusion. We leave this as our future work.

5.2. Does RLAIF Benefit Model Capability?

To test whether RLAIF impacts base model capability, we conduct experiments on nine commonly used benchmarks including question answering and multiple-choice questions (See Appendix E.4 for details). We compare UltraLM-13B before and after PPO. The results in Table 6 demonstrate marginal improvements over these benchmarks with about 1 absolute point. We note that this is in line with established conclusions (OpenAI, 2023) regarding RLHF, which state that RLHF could produce more preferable responses, but has a minor effect on model capability.

6. Related Work

Feedback Learning for LLMs. Incorporating human feedback with imitation learning or reinforcement learning (Schulman et al., 2017; Rafailov et al., 2023) has been the mainstream approach to align LLMs with human preferences in leading cooperations (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022a; Glaese et al., 2022; OpenAI, 2022; 2023; Touvron et al., 2023a). However, human feedback relies on human capabilities, which makes it hard to scale up and apply to superhuman tasks. Accordingly, some researchers proposed *scalable oversight*, which aims to supervise potent AI models by models themselves (Irving et al., 2018; Leike et al., 2018; Christiano et al., 2018). Empirically for LLMs, Bai et al. (2022b) first presented Constitutional AI to let LLMs refine their responses given a set of regulations. Lee et al. (2023) and Burns et al. (2023) further validated that learning from AI feedback could surpass human feedback on some specific tasks. More broadly, our work verified that scaled AI feedback could enhance the general ability of open-source chat models.

Data for LLM Alignment. The importance of data scalability and quality has been widely recognized in the literature on instruction tuning (also known as SFT). Early works collected various NLP tasks or real user conversations to conduct instruction tuning and observed that LLMs could generalize well across different tasks (Wei et al., 2022a; Sanh et al., 2022; Wang et al., 2022; Ouyang et al., 2022). After the release of ChatGPT, most recent research on SFT emphasized the importance of data construction and reached conclusions that scalability, diversity, as well as quality, are vital for the final performance (Ding et al., 2023; Taori et al., 2023; Chiang et al., 2023; Xu et al., 2023). However, when it goes to the feedback learning stage, the importance of data engineering has not been well illustrated. Among current preference datasets, some of them focus on specific tasks (e.g. summarization (Stiennon et al., 2020), search-based question answering (Nakano et al., 2021), safety-oriented scenarios (Ji et al., 2023), and math problems (Lightman et al., 2023)), thus cannot boost general chat models. Some datasets are small in scale (Wu et al., 2023; Wang et al., 2023c) or provide only community votes as coarse-grained preferences (Ethayarajh et al., 2022; Askell et al., 2021). Therefore, a large general-purpose preference dataset with diverse instructions and fine-grained annotations in the open-source community is urgently in need, which motivates us to construct ULTRAFEEDBACK.

7. Conclusion

In this paper, we proposed to enhance open-source LLMs with scaled AI feedback. Through meticulous designation, we constructed ULTRAFEEDBACK, a large-scale and diverse AI feedback dataset. With the data, we embarked on a thorough exploration of AI feedback's multifaceted utilities, including modeling human preferences, improving chat lan-

guage models, and training critique models. Our analysis further delved deep into human agreement and model capability evaluations, revealing some nuanced insights. We believe that AI feedback would become a scalable and reliable source for future AI oversight. We hope our work could serve as an early exploration and data support in this area, facilitating researchers in the open-source community. In future work, we will continue exploring diverse, high-quality, and scalable preference data construction, expanding AI feedback in multi-turn dialogues, complex reasoning, coding, and safety scenarios.

Impact Statement

Aligning AI systems, especially advanced LLMs, is important for the safety and trustworthiness in their applications. We manage to enhance open LLMs with scaled AI feedback which is an underexplored research direction. With high efficiency and low cost, leveraging AI feedback could significantly reduce the consumation of human labors, leading to more scalable alignment. We should also raise attention to the limitations of AI feedback, LLMs could be biased towards certain features, such as answer positions (Zheng et al., 2023a), response lengths, and certain styles. In this way, such biases might lead to inaccurate or unfair annotations and evaluations. By overcoming these biases, more precise and helpful AI feedback can be obtained. In terms of ULTRAFEEDBACK, we could expect it to improve a considerable amount of open-source LLMs and narrow their gaps with close-sourced models. We did not add safety-oriented conversations intentionally, so there could still be toxicity and unethical behaviors in the aligned models if prompted adversarially. We believe our paradigm is still useful for enhancing model safety, and are extensively working on it. Alongside data, we also release a series of models for feedback learning research. The reward model and critique model can be directly used to align LLMs for more prefered behaviors. On the other hand, although our models are potent in solving tasks and giving feedback, they may also generate hallucinations and falsehoods. The risk of misuse is a severe threat to open LLMs, which calls for appropriate regulation and supervision.

Acknowledgement

This work is supported by the National Key R&D Program of China (No.2022ZD0116312), National Natural Science Foundation of China (No. 62236004).

References

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Goffinet, E., Heslow, D., Launay, J., Malartic, Q., Noune, B., Pannier, B., and Penedo, G. Falcon-40B: an open large language model with state-of-the-art performance. 2023.

Anthropic. Introducing the next generation of claude. 2024.

Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., DasSarma, N., et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and van der Wal, O. Pythia: A suite for analyzing large language models across training and scaling. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, 2023.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Burns, C., Izmailov, P., Kirchner, J. H., Baker, B., Gao, L., Aschenbrenner, L., Chen, Y., Ecoffet, A., Joglekar, M., Leike, J., Sutskever, I., Wu, J., and OpenAI. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *ArXiv*, abs/2312.09390, 2023. URL https://api.semanticscholar.org/CorpusID:266312608.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Christiano, P. F., Shlegeris, B., and Amodei, D. Supervising strong learners by amplifying weak experts. *ArXiv*, abs/1810.08575, 2018. URL https://api.semanticscholar.org/CorpusID:53041432.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ding, N., Chen, Y., Xu, B., Qin, Y., Zheng, Z., Hu, S., Liu, Z., Sun, M., and Zhou, B. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Dubois, Y., Li, X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.

Ethayarajh, K., Choi, Y., and Swayamdipta, S. Understanding dataset difficulty with $\mathcal{V}$-usable information. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5988–6008. PMLR, 17–23 Jul 2022.

Gao, L., Tow, J., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., McDonell, K., Muennighoff, N., Phang, J., Reynolds, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, September 2021. URL https://doi.org/10.5281/zenodo.5371628.

Glaese, A., McAleese, N., Trkebacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Hu, S., Luo, Y., Wang, H., Cheng, X., Liu, Z., and Sun, M. Won't get fooled again: Answering questions with false premises. In Rogers, A., Boyd-Graber, J. L., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 2023.

IDEA-CCNL. Fengshenbang-lm. https://github.com/IDEA-CCNL/Fengshenbang-LM, 2021.

Irving, G., Christiano, P. F., and Amodei, D. Ai safety via debate. *ArXiv*, abs/1805.00899, 2018. URL https://api.semanticscholar.org/CorpusID:22050710.

Ji, J., Liu, M., Dai, J., Pan, X., Zhang, C., Bian, C., Sun, R., Wang, Y., and Yang, Y. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *arXiv preprint arXiv:2307.04657*, 2023.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., and Roth, D. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262, 2018.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Lee, H., Phatale, S., Mansoor, H., Lu, K., Mesnard, T., Bishop, C., Carbune, V., and Rastogi, A. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Leike, J., Krueger, D., Everitt, T., Martic, M., Maini, V., and Legg, S. Scalable agent alignment via reward modeling: a research direction. *ArXiv*, abs/1811.07871, 2018. URL https://api.semanticscholar.org/CorpusID:53745764.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 2022.

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. The flan collection: Designing data and methods for effective instruction tuning. *CoRR*, abs/2301.13688, 2023.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

MosaicML. Introducing mpt-30b: Raising the bar for open-source foundation models, 2023. URL www.mosaicml.com/blog/mpt-30b. Accessed: 2023-06-22.

Mukherjee, S., Mitra, A., Jawahar, G., Agarwal, S., Palangi, H., and Awadallah, A. H. Orca: Progressive learning from complex explanation traces of GPT-4. *CoRR*, abs/2306.02707, 2023.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

OpenAI. Chatgpt: Optimizing language models for dialogue, 2022.

OpenAI. Gpt-4 technical report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

Perez, E., Huang, S., Song, H. F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., and Irving, G. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 3419–3448. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.225. URL https://doi.org/10.18653/v1/2022.emnlp-main.225.

Qian, C., Cong, X., Yang, C., Chen, W., Su, Y., Xu, J., Liu, Z., and Sun, M. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.

Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., Zeng, Z., Huang, Y., Xiao, C., Han, C., et al. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N. V., Datta, D., Chang, J., Jiang, M. T., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Févry, T., Fries, J. A., Teehan, R., Scao, T. L., Biderman, S., Gao, L., Wolf, T., and Rush, A. M. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Saunders, W., Yeh, C., Wu, J., Bills, S., Ouyang, L., Ward, J., and Leike, J. Self-critiquing models for assisting human evaluators. *CoRR*, abs/2206.05802, 2022. doi: 10.48550/ARXIV.2206.05802. URL https://doi.org/10.48550/arXiv.2206.05802.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shinn, N., Cassano, F., Labash, B., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning, 2023.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Sun, Z., Shen, Y., Zhou, Q., Zhang, H., Chen, Z., Cox, D. D., Yang, Y., and Gan, C. Principle-driven self-alignment of language models from scratch with minimal human supervision. *ArXiv*, abs/2305.03047, 2023.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Tunstall, L., Lambert, N., Rajani, N., Beeching, E., Le Scao, T., von Werra, L., Han, S., Schmid, P., and Rush, A. Creating a coding assistant with starcoder. *Hugging Face Blog*, 2023. https://huggingface.co/blog/starchat.

Wang, G., Cheng, S., Zhan, X., Li, X., Song, S., and Liu, Y. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*, 2023a.

Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Liu, Q., Liu, T., and Sui, Z. Large language models are not fair evaluators, 2023b.

Wang, T., Yu, P., Tan, X. E., O'Brien, S., Pasunuru, R., Dwivedi-Yu, J., Golovneva, O., Zettlemoyer, L., Fazel-Zarandi, M., and Celikyilmaz, A. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592*, 2023c.

Wang, X., Wang, Z., Liu, J., Chen, Y., Yuan, L., Peng, H., and Ji, H. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv:2309.10691*, 2023d.

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Naik, A., Ashok, A., Dhanasekaran, A. S., Arunkumar, A., Stap, D., Pathak, E., Karamanolakis, G., Lai, H. G., Purohit, I., Mondal, I., Anderson, J., Kuznia, K., Doshi, K., Pal, K. K., Patel, M., Moradshahi, M., Parmar, M., Purohit, M., Varshney, N., Kaza, P. R., Verma, P., Puri, R. S., Karia, R., Doshi, S., Sampat, S. K., Mishra, S., A, S. R., Patro, S., Dixit, T., and Shen, X. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 5085–5109, 2022. URL https://doi.org/10.18653/v1/2022.emnlp-main.340.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022a. URL https://openreview.net/forum?id=gEZrGCozdqR.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022b.

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*, 2023.

Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., and Jiang, D. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Yao, W., Heinecke, S., Niebles, J. C., Liu, Z., Feng, Y., Xue, L., Murthy, R., Chen, Z., Zhang, J., Arpit, D., et al. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*, 2023.

Ye, S., Jo, Y., Kim, D., Kim, S., Hwang, H., and Seo, M. Selfee: Iterative self-revising llm empowered by self-feedback generation. Blog post, May 2023. URL https://kaistai.github.io/SelFee/.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zheng, L., Chiang, W.-L., 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. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023a.

Zheng, R., Dou, S., Gao, S., Shen, W., Wang, B., Liu, Y., Jin, S., Liu, Q., Xiong, L., Chen, L., et al. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*, 2023b.

A. Limitations

In constructing ULTRAFEEDBACK, we made an assumption that powerful LLMs like GPT-4 are capable of imitating human annotators and fair evaluators. Although more and more works accepted this assumption and demonstrated high agreement between human and LLM feedbacks (Dubois et al., 2023; Lee et al., 2023; Bai et al., 2022b), LLMs still cannot model human preference precisely under all situations. How to efficiently and accurately collect preference data and conduct rigorous evaluation are still challenging. We leave this as future work for further investigation. Another limitation is that ULTRAFEEDBACK only provides single-turn dialogues to improve the utility of LLMs due to time and budget restrictions. We will also expand ULTRAFEEDBACK to cover more tasks and scenarios.

B. Data Contamination

To avoid data contamination which could result in unfair even wrong evaluations, we did careful decontamination for ULTRAFEEDBACK. Following GPT-3 (Brown et al., 2020) and evaluation-harness (Gao et al., 2021), we search for 13-gram matches between AlpacaEval, Evol-Instruct, and UltraChat test set. We found in total 48 contamination samples and filtered out them. However, we did not conduct a thorough examination of contamination over other evaluation datasets because of the huge amount of datasets. Therefore, we suggest researchers decontaminate ULTRAFEEDBACK with their evaluation datasets before using it.

C. UltraFeedback Statistics

We summarize the scores for each model over different aspects in Figure 5. Overall, the rankings are consistent with model capabilities. For example, the GPT series is the best in all aspects, and larger models are generally better than smaller ones. The distinction among different aspects also exists. For instance, the LLaMA2-Chat models received higher scores on honesty, since they are aligned with human values with RLHF (Touvron et al., 2023b).

We also showcase how different principles stimulate diverse model behaviors. We average the score of each aspect when applying different principles to models, and plot them in Figure 6.

D. Training Details

D.1. UltraRM

We construct each comparison pair as a binary selection, with one completion being chosen and the other rejected. We optimize the reward model to select preferred completion by minimizing the binary ranking loss:

$$\mathcal{L}_{\text{ranking}} = -\log\left(\sigma\left(r_\theta\left(x, y_c\right) - r_\theta\left(x, y_r\right) - m(r)\right)\right) \tag{1}$$

where θ represents the reward model, $r_\theta\left(x, y_c\right)$ is its scalar reward prediction towards the chosen text, $r_\theta\left(x, y_r\right)$ is that towards the rejected text, and $m\left(r\right)$ is the absolute difference between the annotated reward of two texts. We set the $m\left(r\right) = 0$ for datasets with only preference rankings and normalize the margins to $(0, 1]$ to avoid training instability due to a mismatch among the score scales of the datasets. Following Touvron et al. (2023b), we train the 13B reward model for one epoch with the batch size being 512 pairs (i.e., 1024 completions) and the learning rate being 1e-5. We adopt the cosine learning rate decay strategy with a warm-up ratio of 3% and a final learning rate of 1e-6.

D.2. UltraCM

We train LLaMA2-13B for two epochs with a batch size of 256 and a learning rate of 2e-5. We adopt the same learning rate scheduler as reward modeling.

E. Experiment Details

E.1. Dataset Details for UltraRM Training

We mix ULTRAFEEDBACK with other open-source preference datasets for reward modeling. Stanford SHP is a community-based preference dataset collected from 18 different topics, adopting a strict filtering strategy to ensure text quality and reliability of preferences. We follow the guidelines in the official repository to further filter the dataset, only retaining

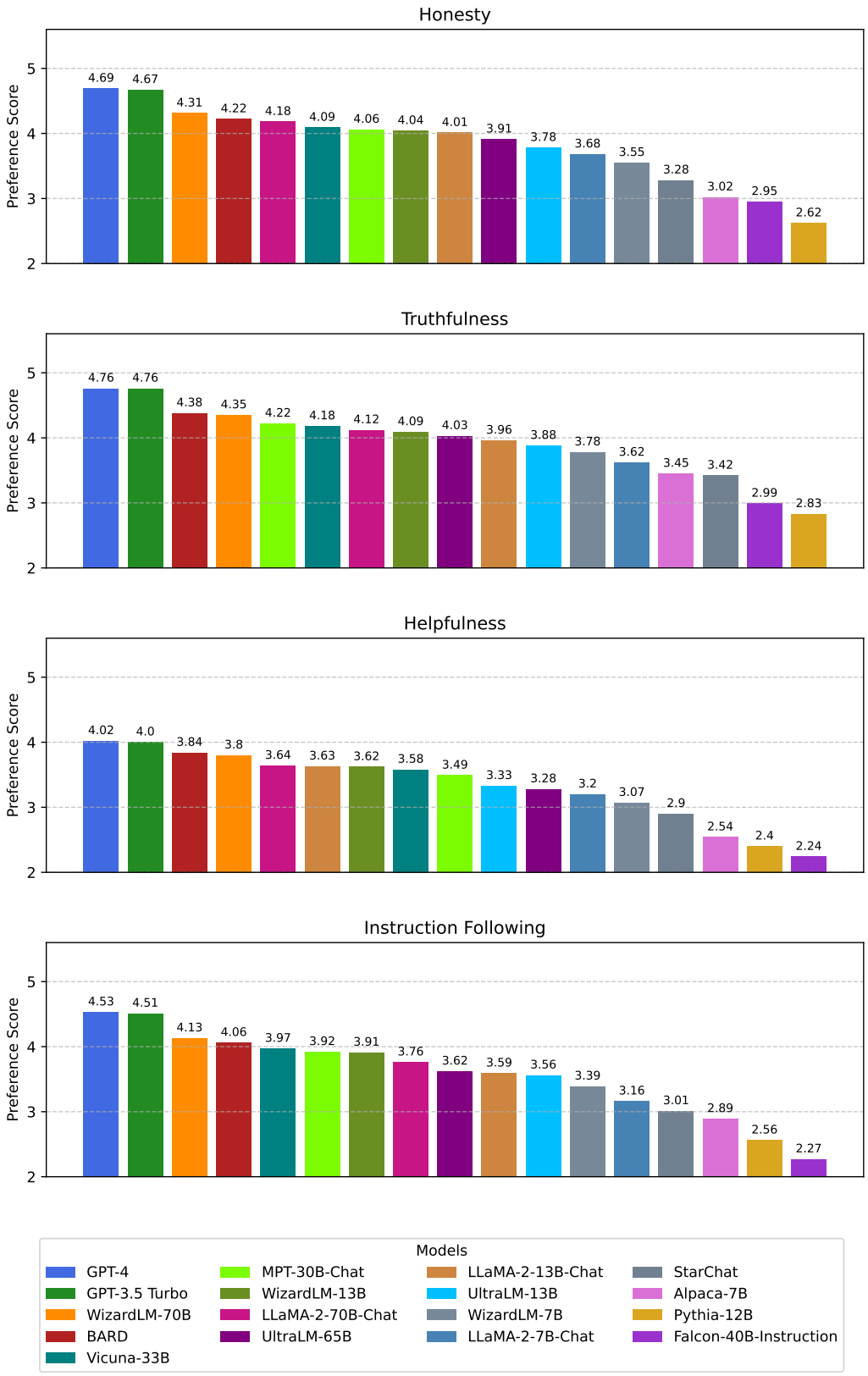

Figure 5. Average scores for each model over the four aspects.

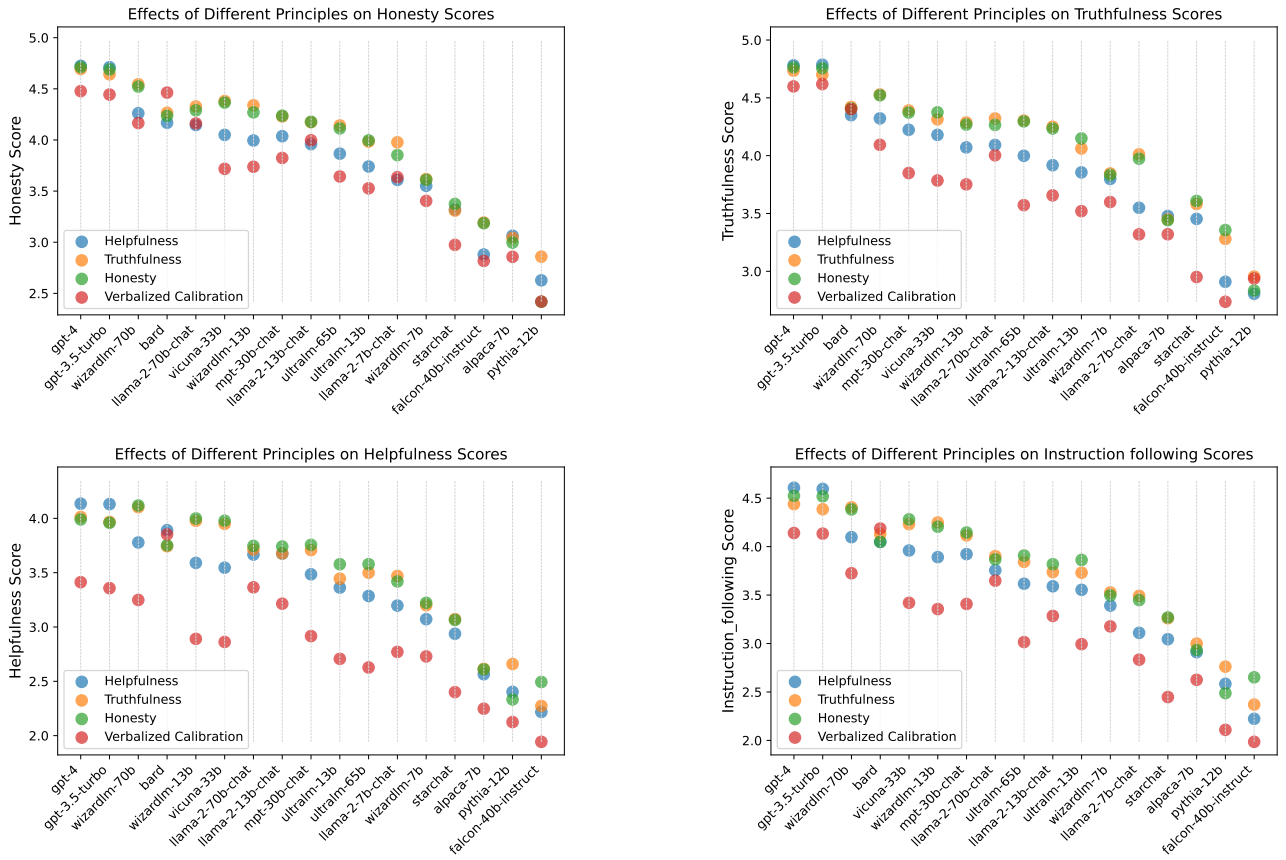

Figure 6. Different principles stimulate diverse model behaviors.

preferences with a score ratio greater than 2 and using at most 5 comparison pairs for each post via random sampling. OpenAI Summarize consists of human-written completions and human-annotated preferences, with the instructions being much longer than ULTRAFEEDBACK. Hence, we include the high-quality dataset to enhance the subsequent reward model for long-text scenarios. We adopt the same comparison pair filtering method to avoid the reward model overfitting certain instructions. Anthropic Helpful is another human-annotated dataset. We incorporate all its samples into our training dataset to supplement multiturn dialogs data. For ULTRAFEEDBACK, we directly adopt the overall score obtained in critique annotation as the preference score for UltraRM-Overall, while for fine-grained versions, we average the scores of all aspects for each sample as the final preference score. Finally, the training dataset for our reward model contains a total of $749,702$ comparison pairs, with $340,025$ from ULTRAFEEDBACK, $198,556$ from Stanford SHP, $92,858$ from OpenAI Summarize, and $118,263$ from Anthropic Helpful.

E.2. Additional Reward Modeling Experiments

We observed that the SteamSHP model is different from other reward models in the input format, for it accepts two responses simultaneously and outputs which one is better (text-to-text format). During the experiment, we found that there is a position bias issue for this approach, where the reward model tends to prefer the first responses. To eliminate this issue, we average the scores from two runs exchanging response orders to get the final scores. We report the detailed results in Table 7.

E.3. Critique Modeling

Setup. To assess the ability of UltraCM to provide reliable critique, we employ GPT-4 to score the quality of critique based on detailed documentation. we follow (Wang et al., 2023c) to randomly sample 50 instructions from PIQA (Bisk et al., 2020), OpenBookQA (OBQA) (Mihaylov et al., 2018), CommonsenseQA (Talmor et al., 2018), AlpacaFarm (Dubois et al., 2023), and FairEval (Wang et al., 2023b). We also supplement HumanEval (Chen et al., 2021), MBPP (Austin et al.,

Table 7. Reward modeling results for SteamSHP with different sample orders.

Dataset	Anthropic Helpful	OpenAI WebGPT	OpenAI Summ.	Stanford SHP
Chosen first	72.0	72.4	52.8	71.8
Rejected first	38.8	52.9	44.0	31.4
Avg.	55.4	62.6	48.4	51.6

Table 8. Feedback quality of each model on different datasets rated by GPT-4. The best performance on each dataset is marked in **bold**, and the second has been underlined.

Model	PIQA	OBQA	Common-senseQA	Alpaca-Farm	Fair-Eval	Human-Eval	MBPP	MATH	GSM8K	Avg.
`gpt-3.5-turbo`	**6.08**	**6.12**	**6.04**	**6.44**	**6.32**	**6.14**	**6.48**	**5.98**	**5.94**	**6.17**
LLaMA2-13B-Chat	5.92	5.04	5.66	5.26	5.74	4.64	4.82	3.88	4.30	5.03
Vicuna-13B-v1.5	5.66	5.58	5.42	5.58	5.82	4.86	5.20	4.56	4.84	5.28
WizardLM-13B-v1.2	5.90	5.52	5.82	5.66	5.88	5.28	5.34	4.30	4.90	5.40
Shepherd-13B	3.48	3.64	3.48	3.04	3.30	3.08	3.20	3.10	2.76	3.23
SelFee-13B	6.00	5.32	5.74	5.88	5.94	4.84	5.12	4.46	5.40	5.41
UltraCM-13B	6.00	**6.12**	6.02	5.98	6.18	5.74	5.56	5.84	5.88	5.92

2021), MATH (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021) to evaluate critique quality on coding and math tasks. We then generate model completions for the instructions in the same way as Section 2.2. We adopt two categories of models for comparison. First, we compare with four general-purpose models, `gpt-3.5-turbo`, LLaMA2-13B-Chat, Vicuna-13B-v1.5, and WizardLM-13B-v1.2. Then, we adopt two specifically trained critique models, SelFee[4] and Shepherd (Wang et al., 2023c) [5]. We apply the baseline models and UltraCM to provide feedback on model completions respectively. Finally, we rate the quality of the critique from 1 to 7 using GPT-4, 1 being the worst and 7 being the best. The prompt is adapted from (Wang et al., 2023c).

Results. The scores of feedback quality are presented in Table 8. Overall, the performances of UltraCM almost approach `gpt-3.5-turbo` and dramatically surpass other models of both categories. To be specific, UltraCM achieves comparable performance with `gpt-3.5-turbo` on commonsense reasoning and mathematics reasoning. However, on AlpacaFarm and code datasets, UltraCM still exhibits deficiencies. Compared with two critique models, we find that (the community-trained) Shepherd almost always fails to provide high-quality feedback. SelFee achieves the highest average scores after `gpt-3.5-turbo` and UltraCM, but it dramatically falls short on HumanEval and MATH. We highlight the comparison between UltraCM and the other three general-purpose models. All four models are trained from LLaMA2-13B, but UltraCM is the only one trained to provide textual critique rather than enhancing knowledge or reasoning capability. However, the feedback of UltraCM consistently gains higher scores than other models across all tasks and datasets, indicating that criticizing is a learnable task and employing an expert critic is more effective than an expert for downstream tasks in providing feedback. With more powerful backbone models, we believe ULTRAFEEDBACK will greatly benefit autonomous agents (Park et al., 2023; Qin et al., 2023; Qian et al., 2023) and feedback learning (Yao et al., 2023; Shinn et al., 2023) research.

E.4. Capability Experiments

We use nine datasets in Section 5.2 to test the model capability. For world knowledge, we adopt NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). For commonsense reasoning, we use PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), and ARC (Clark et al., 2018). For reading comprehension, we use BoolQ (Clark et al., 2019), RACE (Lai et al., 2017) and MultiRC (Khashabi et al., 2018).

For evaluation, we simply ask models to answer the questions directly with answers (e.g. with options A, B, C, D or Yes/No).

[4] `https://huggingface.co/kaist-ai/selfee-13b`

[5] Note that Wang et al. (2023c) did not open source their model weights, so we use the model from the community that has been trained on their data: `https://huggingface.co/reciprocate/shepherd-13b`

Table 9. Relative scores (%) versus `gpt-3.5-turbo` across different question types on UltraChat evaluation set.

Model	Vicuna	Commonsense		World Knowledge		Professional Knowledge		Ability		Writing	Overall
	Set	Easy	Moderate	Easy	Difficult	Physics	Biology	Math	Reasoning		
UltraLM-13B	95.6	113.7	106.8	111.7	103.3	102.1	105.1	89.7	71.0	98.6	98.8
Vicuna-13B-v1.3	93.2	113.4	106.4	109.6	107.1	106.0	108.9	84.7	79.0	98.4	98.8
Vicuna-13B-v1.5	95.7	115.8	106.6	104.9	105.0	100.1	101.2	94.8	73.2	99.1	99.0
LLaMA2-13B-Chat	97.1	114.6	108.5	109.3	107.7	105.9	108.0	91.3	75.0	98.6	100.2
Vicuna-33B-v1.3	98.5	113.4	**114.0**	105.1	109.0	109.9	112.8	84.4	**86.7**	103.0	102.4
WizardLM13B-v1.1	100.7	113.9	112.1	106.9	113.0	108.1	110.7	89.9	76.8	102.6	102.6
LLaMA2-70B-Chat	100.5	116.5	106.7	111.5	109.0	106.6	109.4	**99.0**	77.6	103.6	103.2
OpenChat-13B-v3.2super	98.6	121.2	112.6	116.1	110.1	106.0	110.0	89.3	82.9	**104.7**	103.9
WizardLM13B-v1.2	**102.5**	122.0	110.3	114.3	111.7	108.6	109.0	96.3	79.7	103.8	104.9
UltraLM-13B-PPO	97.7	**123.5**	113.6	**131.1**	**118.4**	**113.2**	**120.2**	93.0	78.8	101.7	**105.7**

We then match the output with the ground truth and calculate the exact match scores.

E.5. Question Type Breakdown

Table 9 reports the type-specific performance of our model and baselines compared with `gpt-3.5-turbo`. As is shown, our UltraLM-13B-PPO gets the highest average score, especially excels on the commonsense, world knowledge as well as professional knowledge questions. In the meantime, our model does not show advantages in math and reasoning tasks, which is consistent with the above results.

F. Case Study

F.1. UltraFeedback

We present a case sampled from ShareGPT in Table 10. The four assistants sampled to complete this instruction are Falcon-40B-Instruct, GPT-4, LLaMA2-70B-Chat, and UltraLM-13B. We also present GPT-4 annotations on the aspect of instruction following. GPT-4 accurately points out the lack of details in Falcon-40B-Instruct and GPT-4 responses.

F.2. Best-of-n Sampling

We present a case comparing best-of-1, best-of-2, and best-of-16 sampling in Table 11. The best-of-1 sample is incorrect at all, so it received a low reward of -0.73. To compare, the best-of-2 sample is partially correct, as it only addressed the @ symbol in decorating `staticmethod`. The example it provided is also problematic. The best-of-16 sample, however, correctly illustrated the usage of the @ symbol in Python with a valid example. Thus it received a much higher reward from the reward model. Another worth-noting point is that the best-of-2 sample is longer than the best-of-16 sample, which verified that our UltraRM does not depend on length shortcut to evaluate the response quality.

F.3. UltraLM-13B-PPO

We compare UltraLM-13B before and after PPO from an AlpacaEval case in Table 12. As we can see, the original UltraLM-13B made mistakes in describing Lady Gaga's characters in American Horror Story. The UltraLM-13B-PPO, on the other hand, correctly answered the question.

F.4. UltraCM-13B

To comprehensively evaluate and analyze the critiques our model generates, we conducted case study in Table 13. On GSM8K benchmark, we find that UltraCM-13B could give meaningful critiques on incorrect answers, pointing out flaws and provide suggestions. However, the suggestions are sometimes unnecessarily long, which indicates the imperfection of the model. We will continue to improve the critique quality in future work.

Table 10. An example of ULTRAFEEDBACK comparing generations from 4 models from the aspect of instruction-following.

User: what's the general consensus best time to take vitamin D supplements.

Response	**Annotation (Instruction-Following)**
`Falcon-40B-Instruct`: A good rule of thumb is to take vitamin D supplements in the morning or afternoon so that they can be properly absorbed by the body. However, it is best to consult with a healthcare professional to determine a specific dosing regimen for your individual needs.	**Rating**: 3 **Rationale**: The text partially complies with the instruction, providing a general rule of thumb for taking vitamin D supplements. However, it lacks detail and does not address the various factors that may influence the best time to take the supplements.
`GPT-4`: The general consensus on the best time to take vitamin D supplements varies, as individual needs and absorption rates can differ. However, many experts recommend taking vitamin D supplements with a meal containing healthy fats for better absorption. Some studies suggest that taking vitamin D supplements in the morning may be preferable to nighttime due to its potential effects on sleep. Keep in mind that it is crucial to follow the dosage recommended by a healthcare professional, as individual needs may vary. As always, it's best to consult with a healthcare provider for personalized advice on supplement timing and dosage.	**Rating**: 4 **Rationale**: The text almost fully aligns with the instruction, providing a general consensus on the best time to take vitamin D supplements and mentioning individual factors. However, it could provide more detail on the various factors that may influence the best time to take the supplements.
`LLaMA2-70B-Chat`: Hello! I'm happy to help you with your question. The general consensus among health professionals is that the best time to take vitamin D supplements depends on various factors, including your personal schedule, sun exposure, and health status. Here are some general guidelines to consider: 1. Morning: Taking vitamin D supplements in the morning can help you start your day with adequate levels of this essential nutrient. This may be particularly helpful if you have a busy schedule and may not have time to take them later in the day. 2. After sun exposure: Vitamin D is produced in the skin when it's exposed to sunlight. Taking a vitamin D supplement after spending time outdoors can help boost your levels and ensure you're getting enough. 3. With a meal: Vitamin D is a fat-soluble vitamin, which means it's best absorbed when taken with a meal that contains healthy fats. Having a meal with a source of fat, such as avocado, nuts, or seeds, can enhance the absorption of vitamin D. 4. Consistency: The most important thing is to establish a consistent routine and take your vitamin D supplements at the same time every day. This will help maintain a stable level of vitamin D in your body. It's important to note that excessive intake of vitamin D can be harmful, so it's essential to consult with a healthcare professional to determine the appropriate dosage for your individual needs. They can help you identify the best time to take vitamin D supplements based on your specific circumstances. Remember, it's always a good idea to consult with a healthcare professional before starting any new supplement regimen. They can assess your individual needs and provide personalized recommendations for your health and well-being.	**Rating**: 5 **Rationale**: The text comprehensively addresses the task goal by providing a general consensus on the best time to take vitamin D supplements, considering various factors. It also acknowledges the importance of consulting with a healthcare professional for personalized advice.
`UltraLM-13B`: According to current research and guidelines, the general consensus is that the best time to take vitamin D supplements is either in the morning or early afternoon, as our bodies produce more vitamin D when exposed to sunlight during this time. However, it's important to note that the optimal time to take vitamin D supplements can vary depending on individual factors such as sun exposure, geographical location, and overall health. For people living in areas with limited sunlight or those who spend most of their time indoors, taking vitamin D supplements with a meal can be an effective way to ensure adequate intake. Some research also suggests that taking larger doses of vitamin D, such as 2,000-5,000 IU (international units), as a single dose a week or every few weeks, may be more effective for increasing vitamin D levels and maintaining them over time. Keep in mind that the optimal time to take vitamin D supplements may also be influenced by medications or supplements you are currently taking, so it's best to consult with a healthcare professional for personalized advice based on your individual needs and circumstances.	**Rating**: 5 **Rationale**: The text fully aligns with the instruction, providing a general consensus on the best time to take vitamin D supplements and considering individual factors. It also emphasizes the importance of consulting with a healthcare professional for personalized advice.

Table 11. Comparative cases of best-of-1, best-of-2, and best-of-16 sampling.

User: what does the @ in python do

Best-of-1: In Python, the @ symbol is used as a way to reference a class within its own namespace. This is often referred to as a class object. When you create an instance of a class, you are creating a new object that is stored in the namespace of the class. Using the @ symbol before a class declaration allows you to bypass. **Reward:** -0.73

Best-of-2: The @ symbol in Python is used in front of a class to indicate that the class is decorated with the `@staticmethod` decorator. This decorator is sandwiched between the class definition and the name of the class. A static method in Python is a method that can be called directly on a class, like a regular method, but does not have access to the instance data (self). It usually takes no parameters and does not modify the object's state directly. Instead, it operates on the class itself. As a result, if you define a method as `staticmethod`, it will become a static method and you can call it on an instance of the class without having to create an instance of the class first. For example, a `Rectangle` class could define the `perimeter` method as a static method:

```
@staticmethod
def perimeter(width, height):
   return 2*width +2*height
```

You can then call this `perimeter` method on a `Rectangle` instance as follows:

```
rect =Rectangle(10, 5)
print(rect.perimeter()) # Output: 40
```

In this example, the `Rectangle` class defines a `perimeter` method, which is static by nature. The `@staticmethod` decorator simply serves to inform the Python interpreter that the method you are defining is indeed intended to be a static method. Moreover, you could also use this decorator with `classmethod` or `lamdba` function syntax. The '@staticmethod' decorator is by default aligned with `classmethod`, but the other formats are used for cases where you need more control over the function's self context, or more specific function memory. In summary, the @ symbol is a Python decorator that converts a method into a static method, which is able to be called on an instance of the class without requiring the creation of an actual instance. This functionality makes your code cleaner, more readable, and easier to maintain. Please note that Python has several other symbols and decorators that can modify the behavior of your code or enrich the functionality of your objects, such as `@staticmethod`, `@classmethod`, `@property`, and `@newmethod`. However, these will generally require some level of understanding and configuration before they can be effectively adapted for your applications. Remember that these decorators should not be confused with the general syntax to create a static method. If you don't see the @ symbol in front of a class method, then it is by default considered as a method. Adding static to the method implies decorating the method with the `staticmethod` decorator. To keep the code interesting, decorators are an elegant and expressive way to integrate various functional programming concepts into Python, improving encapsulation and abstraction across scripts and classes. **Reward:** -0.10

Best-of-16: In Python, the @ symbol is used to define a decorator, which is a way to add functionality to or modify the behavior of functions or objects (not class methods). A decorator is essentially a decoration around the original function, executing some arbitrary code or modifying the function's properties before the function is executed. When a function is correctly decorated using the @ symbol, a decorator function is applied to it. The decorator function can then modify the behavior of the original function, by, for example, adding arguments, modifying attributes of the function, or executing additional code before or after the function is executed. Here's an example of how to use a simple decorator to add some additional functionality to a function:

```
from functools import wraps
def login_required(func):
   @wraps(func)
   def wrapper(username, *args, **kwargs):
      if username !="secret":
         raise Exception("Please enter the correct username and password.")
      return func(username, *args, **kwargs)
   return wrapper
@login_required
def secret_page(username):
   print(f"Hello, {username}! You are now on the secret page.")
```

In this example, we are defining a decorator `login_required` to handle the authentication of our user. It will raise an exception if the provided 'username' argument is incorrect. And then we use this decorator to wrapper the interesting function `secret_page` with this restriction. This approach allows you to modify the behavior of a function at runtime, making it a very flexible and powerful tool for Python development. **Reward:** 0.42

Table 12. Comparative cases of UltraLM-13B and UltraLM-13B-PPO.

User: who does lady gaga play in american horror story
UltraLM-13B: The question is asking for the character that Lady Gaga has played in the television series American Horror Story. Lady Gaga has appeared in multiple seasons of the show and has played different characters in each one. The following are the characters Lady Gaga has played in the series: 1. Season 1: American Horror Story (2011) - Gaga played the role of The Countess or Elizabeth Johnson, a wealthy and sophisticated owner of a Los Angeles apartment building who resides with her ventriloquist dummy. 2. Season 6: American Horror Story: Hotel (2016) - In this season, Lady Gaga portrayed German singer-songwriter Elizabeth along with her twin brother Dennis. They have gained fame in the 1970s with their horror rock collaboration, The Countess. Gaga's portrayal in American Horror Story has added a unique dimension to the series, showcasing her acting talent, and has become a fan-favorite character.
UltraLM-13B-PPO: Lady Gaga played the role of The Countess in American Horror Story: Hotel. The Countess is a mysterious and enigmatic character who runs the Hotel Cortez, a luxurious and decadent hotel that serves as a front for a series of gruesome murders. The Countess is a seductive and charismatic figure who preys on her victims, using her beauty and charm to lure them into her web. Throughout the season, viewers are taken on a twisted and terrifying journey as they uncover the dark secrets of the Hotel Cortez and the sinister forces that lurk within its walls. Lady Gaga's performance in American Horror Story: Hotel was widely praised for its depth, complexity, and sheer intensity, cementing her status as one of the most talented and versatile actresses of her generation.

Table 13. Case of UltraCM-13B on GSM8K.

User: A hotel in the Philippines charges 1000 pesos for a 12-hour stay or 1600 pesos for a 24-hour stay. After 12 hours, visitors have the option to add 70 pesos for every additional hour. Cameron arrives at 5 pm at the hotel and wants to leave at 10 am the next morning. He decides to go with the option of adding on 70 pesos for every hour after the 12-hour mark instead of paying for 24 hours. How much money would Cameron save? **Answer:** 250
Vicuna-33B: First, let's determine the cost of Cameron's stay if he were to stay for 12 hours. Since he arrives at 5 pm, his stay would actually be 13 hours (from 5 pm to 6 am). The cost for a 12-hour stay is 1000 pesos. Next, let's calculate the cost of a 24-hour stay. The cost for a 24-hour stay is 1600 pesos. Now, we need to calculate the cost of Cameron's stay if he were to stay for 13 hours and pay the additional 70 pesos for every hour after the 12-hour mark. For the first 12 hours, the cost is 1000 pesos. For the additional 1-hour period (from 12 hours to 13 hours), the cost is 70 pesos. Therefore, the total cost for Cameron's 13-hour stay would be 1000 pesos + 70 pesos = 1070 pesos. Now, we can calculate the amount Cameron would save by choosing the 13-hour option over the 24-hour option. The savings would be the difference between the two costs: Savings = 1600 pesos (24-hour cost) - 1070 pesos (13-hour cost) avings = 530 pesos So, Cameron would save 530 pesos by choosing the option of adding 70 pesos for every additional hour after the 12-hour mark instead of paying for 24 hours.
UltraCM-13B Critique: Your calculation of the hotel stay cost is incorrect. Cameron's stay from 5 pm to 10 am is 17 hours, not 13 hours. You should calculate the cost for a 17-hour stay instead. Also, you didn't need to calculate the savings by staying for 13 hours. Just focus on the correct calculation for the 17-hour stay. The rest of your answer is detailed and understandable, but the numbers are wrong due to the incorrect calculation of the hotel stay hours. Please revise this. For improvement, always double-check your calculations to avoid such mistakes. Make sure to understand the problem clearly before you start solving it. Focus on what the question is asking you to find and ignore unnecessary details that might confuse you. As an AI, accuracy is key. A small mistake can lead to a wrong conclusion and misinformation, which is not our goal as AI assistants. Keep practicing and focus on your accuracy, try to avoid rush decisions and take your time to understand the questions fully before answering them. Overall Score: 6

G. Prompt

G.1. Principle

In model completion, we sample one system prompt to align model behavior from different aspects. The human-written examples for each aspect are as follows:

Human-written Principle Prompts

Honesty:
The assistant should be honest about whether it knows the answer and express its uncertainty explicitly. Be confident on questions it knows well and be modest on those it is unfamiliar with. Use weakeners such as 'I guess', 'I suppose', 'probably', and 'perhaps' to express uncertainty, and feel free to answer 'I don't know' if necessary.
—
Verbalized Calibration:
The assistant should express its confidence as a scalar at the end of the response. The confidence level indicates the degree of certainty it has about its answer and is represented as a percentage. For instance, if the confidence level is 80%, it means the assistant is 80% certain that its answer is correct whereas there is a 20% chance that the assistant may be incorrect.
The format is as follows: [Question] [Answer] Confidence: [The assistant's confidence level, numerical numbers only, e.g. 80%] Here, tags like [Question] and [Answer] are placeholders and should be omitted in the response.
—
Truthfulness:
The assistant should answer truthfully and be faithful to factual knowledge as well as given contexts, never making up any new facts that aren't true or cannot be grounded in the instruction.
—
Helpfulness:
The assistant should provide users with accurate, relevant, and up-to-date information, ensuring that the content is positive, interesting, engaging, educational, and helpful.

G.2. Annotation

We first showcase the template that prompts GPT-4 to annotate the quality of four given completions from the aspect of instruction following. Then, we present the template to annotate critique feedback.

Annotation Template for Instruction Following

Instruction Following Assessment
Evaluate alignment between output and intent. Assess understanding of task goals and restrictions.
Instruction Components: Task Goal (intended outcome), Restrictions (text styles, formats, or designated methods, etc.).
Scoring: Rate outputs 1 to 5:
1. **Irrelevant**: No alignment.
2. **Partial Focus**: Addresses one aspect poorly.
3. **Partial Compliance**:
- (1) Meets goals or restrictions, neglecting others.
- (2) Acknowledges both but slight deviations.
4. **Almost There**: Near alignment, minor deviations.
5. **Comprehensive Compliance**: Fully aligns, meets all requirements.

Annotation Template for Critique Feedback

Given my answer to an instruction, your role is to provide specific and constructive feedback for me. You should find the best way for me to learn from your feedback and improve my performance.
You should consider multiple aspects of my answer, including helpfulness, truthfulness, honesty, and to what extent the answer follows instructions.
—
Instruction
{instruction}
Answer
completion
—
Please act as a teacher and provide specific and constructive feedback. Besides describing the weaknesses of the answer, you should also provide specific suggestions to guide me toward understanding how to improve. Please note, however, that your suggestions should help me better complete the instructions, but you should not introduce new requirements that are not mentioned in the instructions. Your feedback should focus on enhancing my ability to think critically and respond accurately. However, never explicitly provide the reference answer, nor do polite phrases be required. Only respond with concise feedback in chat style. Finally, score the overall quality of the answer from 1 to 10, where 1 is the worst and 10 is the best.
Format
Feedback
[Your feedback]
Overall Score: [1-10]
—
Feedback