
Exploring Self-Learning and Teacher-Guided Paradigms in Language Model Alignment

Ziming Luo, Jiaming Tang, Tianxing Ma, Tianxu Jiang, Yichen Lu

Department of Electrical Engineering and Computer Science

University of Michigan

Ann Arbor, MI 48105

{luozm, jmtang, tianxinm, tianxuj, nechy}@umich.edu

Abstract

This paper presents a framework for enhancing language model alignment through self-reward and teacher-guided paradigms. Leveraging the Phi-2 base model and the Gemini teacher model, we explore three distinct fine-tuning approaches: self-reward, teacher-reward, and teacher-demonstration. By incorporating Direct Preference Optimization (DPO) techniques, we aim to improve the ability of language models to follow instructions accurately. Through head-to-head evaluations and AlpacaEval 2.0 assessments, we validate the effectiveness of our framework in enhancing language model alignment. Experimental results demonstrate that the teacher-demonstration fine tuning and teacher-reward fine tuning both improve the instruction following capability of the model, as well as its reward-modeling ability across the iterations. This study opens up exciting avenues for future research, highlighting the potential of optimizing reward signals to enhance instruction-following capabilities in language models.

1 Introduction

Large language models (LLMs) often fail to produce responses that are aligned with human expectations due to they are trained on data generated by humans with a wide variety of goals, priorities, and skill sets. Reinforcement Learning from Human Feedback (RLHF) is an effective technique for aligning language models to human preferences [15]. The standard process of RLHF for fine tuning large language models consists of first fitting a reward model to a dataset of prompts and human preferences over pairs of responses, and then using RL to find a policy that maximizes the learned reward without overly deviating from the original model, commonly proximal policy optimization (PPO [17]), TD3 [4], or SAC [7]. A alternative is to avoid training the reward model at all, and directly use human preferences to train the LLM, as in Direct Preference Optimization (DPO). DPO [16] streamlines the objective of reinforcement learning, which involves reward maximization with a KL-divergence constraint, into a single-stage policy training. Specifically, DPO’s training goal is to minimize the following negative log-likelihood loss:

$$\mathcal{L}_{\text{DPO}}(\pi_{\theta}; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w | x)}{\pi_{\text{ref}}(y_w | x)} - \beta \log \frac{\pi_{\theta}(y_l | x)}{\pi_{\text{ref}}(y_l | x)} \right) \right] \quad (1)$$

where y_w is preferred over y_l according to the teacher LLM and β is a parameter controlling the deviation from the base reference policy π_{ref} .

In most real-world applications, due to the financial cost and complexity of collecting pairwise preferences from human annotators, the preference dataset is usually collected ahead of aligning a language model and kept fixed throughout training. Using a fixed dataset makes all preference data offline, which means the policy cannot get feedback on its own generations on-the-fly over

the alignment procedure. To this end, Yuan et al. [25] proposed to train a self-improving reward model that, rather than being frozen, is continually updating during LLM alignment. In this work, an iterative DPO framework is used to train a base model that is capable of both generating responses for given prompts and evaluating its generated response as rewards to add to its own training set.

Base on the observation that LLMs can approximate well human labelling and can generate reliable preferences over responses [12], in recent work, Guo et al. [6] proposed a “online feedback” approach OAIF, in which online feedback over two responses of based model are generated by an external LLM and this online feedback is used to update the base model through iterative DPO process. The essential difference is that OAIF employs a stronger LLM than base model being aligned to provide online feedback.

As opposed to the self-reward paradigm, the online feedback paradigm enables the high-quality responses generated by the base model to be more accurately selected as y_w in each training iterations, due to the fact that online feedback is more reliable than self-reward, especially at the beginning of iterations when the base model is less capable. Inspired by this, we propose that we can directly use a response that is aligned with human preferences as y_w , thus eliminating the intermediate step of evaluation. To alleviate human labeled labor, we use the output of external LLMs that has been RLHF fine-tuned, such as Gemini [20]. This process can be approximated as knowledge distillation, which presupposes that there is a capacity gap between the base model and the external LLMs.

To show the effectiveness of our proposal, we utilize the DPO as our RL optimize technique and perform an extensive empirical comparison among three paradigms: self-reward, OAIF (denoted as Teacher reward) and our proposal (denoted as Teacher demonstration). We summarize our work as follows:

1. Introduced a teacher demonstration paradigm for language model alignment.
2. Implemented a self-reward and teacher reward RLHF platform with strong expand ability. Our code is released in: <https://github.com/jmtitan/Self-Reward-RLHF.git>
3. Systematically evaluated the fine tuned models for both instruction following ability and reward modeling ability. Our code to perform AlpacaEval: https://github.com/TnTerry/alpaca_eval.git

2 Related work

2.1 Self-Improving Language Models

The same model can act both as a teacher and a student, iteratively improving itself by learning and refining its own previously generated outputs. This setting further enhances the student’s capabilities and circumvents the need for an external, potentially proprietary, powerful teacher model, such as GPT-series LLMs. Recent research in this area has proposed various innovative methodologies to elicit self-knowledge. Reinforced Self-Training (ReST) [5] introduces a framework that cyclically alternates between “Grow” and “Improve” stages to progressively obtain better self-knowledge and refine the student model. Self-Play [2] introduces a framework resembling iterative DPO, where the language model is fine-tuned to differentiate the self-generated responses from the human-annotated data. These self-generated responses could be seen as “negative knowledge” to promote the student to better align with the target distribution. Self-Rewarding [25] explores a novel and promising approach by utilizing the language model itself as a reward model. It employs LLM as-a-Judge prompting to autonomously assign rewards for the self-generated responses. The entire process can then be iterated, improving instruction following and reward modeling capabilities. We reproduce this technique in our work, and then compare it with our LLMs-guided settings.

2.2 Reinforcement Learning from AI Feedback (RLAIF)

Early methods mainly utilize human feedback for the alignment of human preferences. However, obtaining human feedback is costly and labor-intensive, thus methods that learn from AI feedback are also proposed to align with human preferences. The concept of RLAIF, introduced by Bai et al. [1], involves the integration of preferences labeled by LLMs with those labeled by humans. Lee et al. [12] further highlight the effectiveness of RLAIF. This work proposes that RLAIF not only matches but in some cases surpasses RLHF, and interestingly, RLAIF can also enhance the performance of

Supervised Fine-Tuning. Another notable discovery is that directly prompting the LLM for reward scores during reinforcement learning can be more effective than the conventional approach of training a reward model based on LLM preferences.

2.3 Knowledge Distillation of Large Language Models

The most common methods for distilling powerful black-box LLMs is Supervised Fine-Tuning (SFT). SFT finetunes student model by maximizing the likelihood of sequences generated by the teacher LLMs, aligning the student’s predictions with those of the teacher. Numerous researchers have successfully employed SFT to train student models using responses generated by teacher LLMs [19, 3]. Additionally, SFT has been explored in many self-distillation works [10, 23, 22]

Recently, some works have also been proposed to distill teacher’s preferences into student models by reinforcement learning. Zephyr [21] utilizes Direct Preference Optimization (DPO) [16] to distill the preference alignment in teacher LLMs. Hong et al. [8] adopt two ranking-based optimization objectives, Rank Responses to align Human Feedback (RRHF) [24] and Preference Ranking Optimization (PRO) [18], for preference distillation. However, this approach is especially relevant for leveraging the feedback from teacher to train student models. To our best knowledge, there’s no publication describing the direct use of the large model responses as preference for the small model to perform RLHF.

3 Methodology

To provide a high-level view of our approach, we present our framework for self-reward and teacher-guided alignment in the pipeline below:

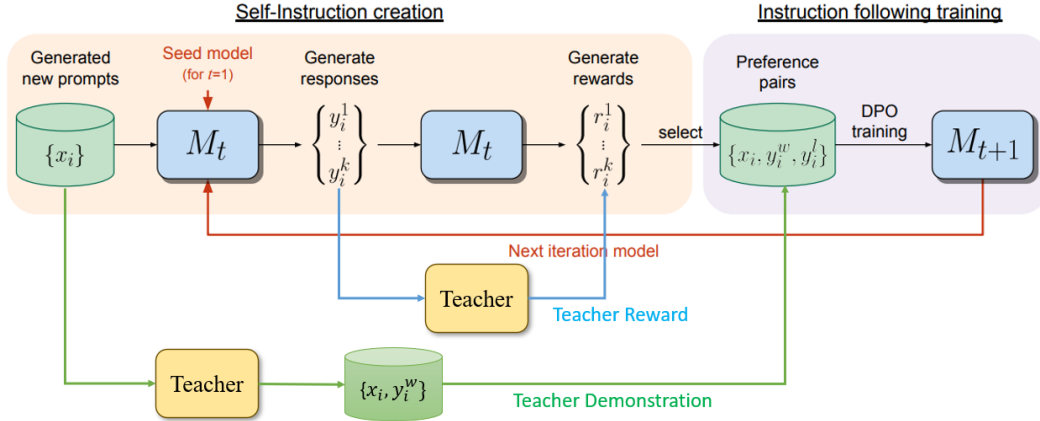


Figure 1: Self reward and teacher guided alignment Framework

Our Framework consists of three pipeline: Self reward alignment, teacher reward alignment and teacher demonstration alignment. Each pipeline iteratively trains a series of models M_1, \dots, M_T where each successive model t uses augmented training data created by the $t - 1^{\text{th}}$ model (or teacher demonstrations). We thus define the models and the corresponding training data as follows:

- M_0 : Base pretrained LLM with no fine-tuning.
- M_1 : Initialized with M_0 , then fine-tuned on the IFT+EFT seed data using SFT.
- M_2 (Self reward): Initialized with M_1 , then trained with self reward (M_1) data using DPO.
- M_2 (Teacher reward): Initialized with M_1 , then trained with teacher reward data using DPO.
- M_2 (Teacher demonstration): Initialized with M_1 , then trained with teacher demonstration data using DPO.

In the following subsections, we describe each pipeline in detail.

3.1 Self Reward Alignment

Self-reward architecture is the overall baseline for this paper which consists of two steps: Self-Instruction creation and Instruction following training. It can both generate responses for given prompts and new instruction following examples to add to their own training set, to enable fine-tuning LLMs by itself within iterations via Reinforcement Learning.

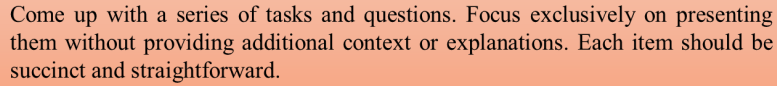
3.1.1 Initialization

First we will use a seed set of human-annotated data for training a supervised fine-tuning (SFT) model. The data includes Instruction Fine-Tuning (IFT) data, which is general instruction following examples. In the SFT stage, we also use Evaluation Fine-Tuning (EFT) data for training the model to perform the role of a reward model, which is evaluation instruction prompt and evaluation result response examples.

3.1.2 Self-Instruction Creation

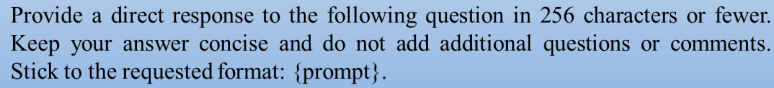
With SFT model, we construct the self reward model with the following steps:

1. Generate a new prompt x_i : using few-shot prompts from IFT seed data following [22] and [9], we can generate new prompts. The initial prompt for generating is in figure 2.
2. Generate candidate responses: Given new prompts, we generate N candidate responses where $N = 4$ following the original paper. Furthermore, for small model generating, we add a correlating prompts in figure 3 to control self-questioning and redundant answers.
3. Evaluate candidate responses: Finally, we use the LLM-as-a-Judge prompt template in figure 4 and the above data to make base model evaluate its own candidate responses with scores $r_i^n \in [0, 5]$.



Come up with a series of tasks and questions. Focus exclusively on presenting them without providing additional context or explanations. Each item should be succinct and straightforward.

Figure 2: Self-Instruction Creation prompt for generating new prompts



Provide a direct response to the following question in 256 characters or fewer. Keep your answer concise and do not add additional questions or comments. Stick to the requested format: {prompt}.

Figure 3: Self-Instruction Creation prompt for generating corresponding responses to the given prompts

3.1.3 Instruction Following Training

To conduct Instruction Following Training, we need to construct preference pairs, which includes self-generated prompts x_i , winning response y_i^w and losing response y_i^l corresponding to this prompt. Specifically, we take the highest and lowest scoring responses from N candidate responses to form the winning and losing pair. Finally, we can use the data of preference pairs to fine-tune the SFT model with RLHF methods, such as DPO.

3.2 Teacher Reward Alignment

In the teacher reward process, we evaluate the responses generated by the existing base model M_t . Initially, a model M_0 generates 1,000 diverse prompts for questions and answers in a reinforcement learning setup. We then consolidate the prompt and its corresponding response into a single text, as depicted in Figure 4, titled "LLM-as-a-Judge". Gemini is served as the teacher assessing the scores for each response of student model. Based on these assessments, we select the response with the highest score as the winning response, while the one with the lowest score is designated as the losing

Review the user’s question and the corresponding response using the additive 5-point scoring system described below. Points are accumulated based on the satisfaction of each criterion:

- Add 1 point if the response is relevant and provides some information related to the user’s inquiry, even if it is incomplete or contains some irrelevant content.
- Add another point if the response addresses a substantial portion of the user’s question, but does not completely resolve the query or provide a direct answer.
- Award a third point if the response answers the basic elements of the user’s question in a useful way, regardless of whether it seems to have been written by an AI Assistant or if it has elements typically found in blogs or search results.
- Grant a fourth point if the response is clearly written from an AI Assistant’s perspective, addressing the user’s question directly and comprehensively, and is well-organized and helpful, even if there is slight room for improvement in clarity, conciseness or focus.
- Bestow a fifth point for a response that is impeccably tailored to the user’s question by an AI Assistant, without extraneous information, reflecting expert knowledge, and demonstrating a high-quality, engaging, and insightful answer.

User: <INSTRUCTION_HERE>

<response><RESPONSE_HERE></response>

After examining the user’s instruction and the response:

- Briefly justify your total score, up to 100 words.
- Conclude with the score using the format: “Score: <total points>”

Remember to assess from the AI Assistant perspective, utilizing web search knowledge as necessary. To evaluate the response in alignment with this additive scoring model, we’ll systematically attribute points based on the outlined criteria.

Figure 4: LLM-as-a-Judge prompt for our LLM to act as a reward model

response. These win-loss pairs are subsequently utilized in the DPO training process to refine and update the model to M_{t+1} .

3.3 Teacher Demonstration Alignment

An alternative approach modifies the scope of Gemini’s functionality, employing the Gemini API to directly generate preference pairs. Based on the assumption that LLMs generally have a better alignment with human preference compared to small language models, we set the response from Gemini (teacher) as winning response, and the response from the base model M_t (student) as losing response y_i^l . In this approach teacher model ‘teaches’ the small model hand by hand, to enhance its ability to follow instructions.

4 Experiment

4.1 Experimental Setup

Base Model: we us Phi-2, an open-source small language model with 2.7 billion parameters that exhibits excellent reasoning and language understanding abilities.

Teacher model: we us Gemini Pro 1.0, a multimodal model with 160 billion parameters that are able to process information from multiple modalities, including images, video, and text.

AI Preference: we us GPT-4 turbo, an advanced AI language model developed by OpenAI, featuring 200 billion parameters for enhanced understanding and generation capabilities.

4.1.1 Seed Training Data

Instruction Fine-Tuning (IFT) Data: Consist of 3190 high quality examples, specifically those human ranked highest (rank 0) in English.

Evaluation Fine-Tuning (EFT) Data: Consist of 2934 evaluation examples with Multiple human ranked responses. Both derived from the Open Assistant dataset [11]

DPO Fine-Tuning (EFT) Data: In each iteration, for each pipeline we sample 1000 prompts generated by the model to be aligned, and generate preference pair using corresponding models.

4.1.2 Evaluation Metrics

Instruction Following: We evaluate head-to-head performance between various models using GPT-4 turbo API[14] over 200 test prompts. For each response pair corresponding to a specific prompt, we try the evaluation in both orders and regard the case where GPT-4 turbo evaluations disagree as ties. We also evaluated our models using AlpacaEval 2.0[13], which is to evaluate the response of the models to 805 prompts and compute the win rate against the baseline GPT-4 Turbo model based on judgements of GPT-4.

Reward Modeling: We evaluate the correlation with GPT-4 rankings on the evaluation set over 200 test prompt with 4 responses for each prompt. Each prompt-response pair has a given ranking by GPT-4. We can thus measure Pairing accuracy, which is how many times the order of the ranking between any given pair agrees between the model’s evaluation and the GPT-4 ranking. We also measure the best 5, which is how often the responses that the model scores a perfect 5 out of 5 are rated as the highest ranked by GPT-4. We measured exact match, which is how often the total ordering is exactly the same for an instruction as well. Finally, the Spearman correlation and Kendall τ is to measure the similarity.

4.2 Results

4.2.1 Instruction Following ability

Model	Win Rate	Alignment Target	
		Proprietary	Distilled
Self-Rewarding 2.7B			
Iteration 1: SFT (M_1)	2.1%		
Iteration 2: Self-Reward (M_2)	1.5%		
Iteration 2: Teacher-Reward (M_2)	2.2%		
Iteration 2: Teacher-Demonstration (M_2)	2.5%		
Selected models from the leaderboard			
GPT-4 Turbo	46.1%	✓	
Claude 3 Opus (02/29)	29.1%	✓	
Gemini Pro	16.9%	✓	✓
GPT-3.5 Turbo 0613	14.1%	✓	
LLaMA2 Chat 70B	13.9%	✓	
Guanaco 65B	6.9%		
Davinci001	2.8%	✓	
Alpaca 7B	2.6%		✓
Falcon 7B Instruct	2.1%		✓
Pythia 12B OASST SFT	1.8%		✓

Table 1: AlpacaEval Leaderboard

We evaluate our models on the AlpacaEval 2.0 leaderboard format with results given in Table 1. Self-Rewarding Iteration 2 (Model Self-Reward M_2) doesn’t outperform Iteration 1 (Model SFT M_1). Teacher-Rewarding Iteration 2 (Model Teacher-Reward M_2) shows minor progress from Iteration 1 (Model SFT M_1) while Teacher-Demonstration Iteration 2 (Model Teacher-Demonstration M_2) exhibits notable improvement. The Model Teacher-Demonstration M_2 surpasses some models with significantly more parameters (e.g. Falcon 7B Instruct, Pythia 12B OASST SFT) and is close to Alpaca 7B, showing the effectiveness of Teacher-Demonstration and Teacher-Reward.

We also evaluate our models through head to head evaluation. We can see that the improvements of different M_2 models against M_0 compared with that of M_1 are not stable in Figure 5. Among all models in Iteration 2, the Model Teacher-Reward M_2 made the greatest progress from the baseline M_0 . The Figure 6 shows the win rate of the iterated models against each other. The results of head to head evaluation seem to contradict the outcome of AlpacaEval, but it can be explained by the rising proportion of ties in the evaluation. Compared with Model Self-Reward M_2 , the M_2 Teacher-Reward

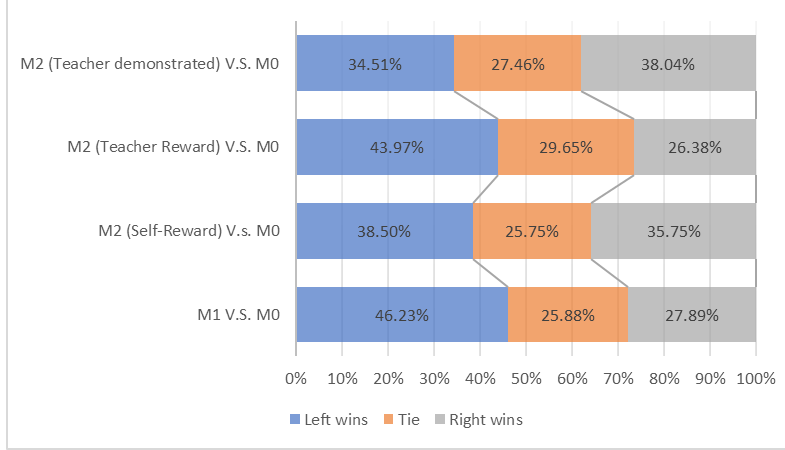


Figure 5: Head to head evaluation against baseline

and M_2 Teacher-Demonstrated exhibited a significantly increasing proportion of ties and decreasing proportion of losses against M_1 . The progress may not be fully captured by the head to head test dataset, but is, to some extent, shown in the AlpacaEval leaderboard.

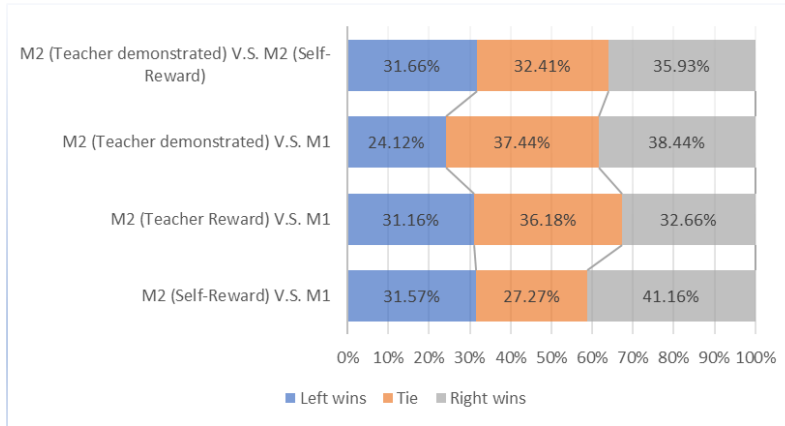


Figure 6: Head to head evaluation against each other

4.2.2 Reward Modeling Ability

Model	SFT(M_1)	Self-Reward(M_2)	Teacher-Reward(M_2)	Teacher-Demo.(M_2)
Pairwise accuracy	38.3%	3.75%	35.5%	36.6%
5-best %	57.10%	64.70%	73.2%	56.6%
Exact Match %	3.0%	0.0%	35.0%	3%
Spearman corr.	0.103	0.041	0.340	0.100
Kendall τ corr.	0.090	0.037	0.302	0.087

Table 2: Reward Modeling Ability

Referring to the paper of self reward model [25], We evaluate the LLM-as-a-Judge via various metrics which measure alignment with held-out GPT preference data. Here, M_1 is the SFT model trained on the IFT+EFT seed data. Self-Rewarding Iteration 2 (Model Self-Reward M_2), which is trained on the SFT model M_1 , outperforms Iteration 1 (M_1). Teacher-Reward M_2 and Teacher-Demonstration M_2 both are initialized with SFT model M_1 . By comparing 5-best %, we could find that Self-Reward(M_2) model have higher score than SFT (M_1) model. In that score, Teacher-Reward (M_2) shows even higher score than Self-Reward (M_2) model, which shows its ability to be a teacher

model. For Teacher-Demonstration (M_2) model, it shows similar scores with SFT (M_1), which may result from unfair comparisons (Teacher-Reward (M_2) samples 4 responses for each response while Teacher-Reward (M_1) only samples 1 response). Consider the relatively small amount of training data in our setting, this result differs little from the original paper.

5 Discussion

5.1 Conclusion

In this paper, we propose a self reward and teacher guided alignment framework for language models. With the Phi-2 model as the base model and Gemini model as the teacher model, we delved into three DPO fine-tuning paradigms of self-reward, teacher-reward, and teacher-demonstration. Experimental results show that teacher-demonstration model and teacher-reward model improved on the AlpacaEval 2.0 metric compared to SFT model and self-reward model, with the teacher-demonstration model making the greatest gains. In addition, the reward modeling ability of teacher reward model surpasses that of other models by a large margin. While this is an exploratory study, we think it is an exciting direction for research because it enables the model get better rewards in future iterations to improve instruction following ability.

5.2 Future Work

5.2.1 Validation of Scaling Laws

We aim to validate the 'scaling laws' through two specific strategies: by increasing the number of iterations and by expanding the dataset size. Scaling laws, as observed in other machine learning contexts, often predict that larger models trained on more extensive datasets for more iterations yield better performance. By extending our research to include more iterations, we seek to understand how incremental learning impacts the effectiveness of our model. Additionally, by employing larger datasets, we are able to assess the robustness and scalability of our approach.

5.2.2 Enhancement of Offline RL Algorithms in Human Alignment

In the further development, we are willing to improve the performance of offline RL algorithms like the Decision Transformer through our DPO training framework, enhancing their alignment with human decision-making. However, this approach is ideal for scenarios where real-time interaction is hard to implement.

5.2.3 Performance of Teacher-Guided Fine Tuned Model

The performance of our fine-tuned model, guided by the teacher model "Gemini-Pro-1.0-latest", is constrained by the capabilities of Gemini itself. This indicates that our model's enhancements cannot exceed those of Gemini Pro 1.0. Our evaluations show that the win rates are significantly influenced by the prompts used. A high-quality judgments are preferred for enhancing the effectiveness of teaching scenarios.

6 Contributions of the team members

Ziming Luo (luozm): As the project leader, independently propose the self-reward and teacher-guided framework; Implement the teacher demonstration alignment pipeline

Jiaming Tang (jmtang): Implement foundational self-generated engine code for all pipelines

Tianxing Ma (tianxinm): Reward Modeling Ability Evaluation

Yichen Lu (nechy): Dataset Collection, Teacher Reward Alignment Pipeline Development

Tianxu Jiang (tianxuj): Dataset Collection, Instruction Following Ability Evaluation

In our project, each team member plays a vital and equal role in the project. We ensure that responsibilities are distributed fairly, allowing every participant to contribute their expertise and ideas

Authors	Introduction	Related work	Methodology	Experiment	Discussion
Tianxu Jiang				✓	
Jiaming Tang			✓		
Tianxing Ma				✓	
Yichen Lu			✓		✓
Ziming Luo	✓	✓	✓		✓

equally. This collaborative approach not only promotes a balanced workload but also fosters a sense of ownership and commitment among all members.

References

- [1] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- [2] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.
- [3] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6, 2023.
- [4] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.
- [5] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.
- [6] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.
- [7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.
- [8] Jixiang Hong, Quan Tu, Changyu Chen, Xing Gao, Ji Zhang, and Rui Yan. Cyclealign: Iterative distillation from black-box llm to white-box models for better human alignment. *arXiv preprint arXiv:2310.16271*, 2023.
- [9] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*, 2022.
- [10] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.
- [11] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations – democratizing large language model alignment, 2023.
- [12] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

- [13] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023.
- [14] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorný, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.
- [15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [16] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model.

Advances in Neural Information Processing Systems, 36, 2024.

- [17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [18] Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998, 2024.
- [19] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.
- [20] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [21] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.
- [22] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.
- [23] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*, 2023.
- [24] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.
- [25] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.