SAIL: Self-improving Efficient Online Alignment of Large Language Models

Vibhu Agrawal

1 Introduction

What is the problem? AI alignment has become a crucial part of large language model (LLM) training and has allowed LLMs to demonstrate improved performance in a variety of tasks. Reinforcement Learning from Human Feedback (RLHF) has been a crucial component of this technique because of its effectiveness. RLHF primarily operates in three steps: (Step 1) supervised fine-tuning, (Step 2) reward learning, and (Step 3) policy optimization. Recent research by Rafailov et al. (2023) has shown that reward learning can be implicitly learned through a new formulation, Direct Preference Optimization (DPO). However, both these and other methods (Agarwal et al., 2020; Ouyang et al., 2022; Chakraborty et al., 2024; Swamy et al., 2024) have used static datasets which can introduce issues in generalization because of unseen data in practical settings.

Online RLHF (Guo et al., 2024; Sharma et al., 2024; Lee et al., 2023; Yuan et al., 2024) has been the alternative to static datasets, and typically works by allowing an LLM to generate new responses during fine-tuning in successive iterations. However current research does not address a key problem, **the interdependence of model and data**. When LLMs are used to generate the preference or response data that is used to fine-tune in later iterations in online RLHF, it leads to distribution shifts because the LLM learns from its own generated data which is suboptimal to ground-truth data generated by humans. This is not a problem for offline RLHF, where datasets are often created and annotated by humans.

Why is this problem important? This issue of model-data interdependence in online RLHF is critical because it fundamentally affects the robustness and reliability of LLMs in real-world applications. When models train on self-generated data, there is a risk of compounding errors or amplifying biases inherent in earlier iterations. Over successive fine-tuning cycles, these distribution shifts can result in performance degradation, making the model less capable of handling diverse or unexpected inputs. In high-stakes applications such as medical diagnosis, legal decision-making, or autonomous systems, this lack of robustness could lead to significant consequences, including incorrect predictions or unethical behavior. Addressing this challenge is essential to ensure that LLMs can generalize effectively and maintain alignment with human intentions, even in dynamic and complex operational settings. Resolving this issue could pave the way for more scalable and reliable methods for deploying LLMs in environments where access to high-quality human-annotated data is limited.

2 Our Approach

Together with my coauthors, Mucong Ding, Souradip Chakraborty, Zora Che, Alec Koppel, Mengdi Wang, Amrit Bedi, and Furong Huang, we propose the following solution to the problem of resolving the interdependence of model and data in online RLHF.

Previous work (Chakraborty et al., 2023; Shen et al., 2024) has proposed a bilevel formulation of Direct Preference Optimization to effectively capture the relationship between reward learning and policy optimization. Our solution improves on this existing formulation by transforming the bilevel formulation into a single-level form. This allows us to solve the formulation in an efficient manner as compared to typical methods for bilevel optimization.

Mathematical Notations. We define the language model formally. Let the vocabulary set be $\mathcal{V}$ and the language model be represented as a mapping π. The model takes a sequence of tokens (the prompt) $\mathbf{x} := \{x_1, x_2, \cdots, x_N\}$, with the set of all prompts denoted by $\mathcal{P}$, and generates a response $\mathbf{y} := \{y_1, y_2, \cdots, y_T\}$ token by token. At each time step t, the model receives the input prompt x

combined with the generated tokens up to $t-1$, denoted as $[\mathbf{x}, \mathbf{y}_{<t}]$.The next token y_t is then sampled according to $y_t \sim \pi(\cdot \mid [\mathbf{x}, \mathbf{y}_{<t}])$

To give some context, the reward learning in RLHF is typically represented as

$$\mathcal{L}_R(r, \mathcal{D}_r) = -\mathbb{E}_{(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) \sim \mathcal{D}_r} \Big[\log \sigma \big(r(\mathbf{x}, \mathbf{y}_w) - r(\mathbf{x}, \mathbf{y}_l) \big) \Big], \tag{1}$$

where $\mathcal{D}_r$ represents the dataset of responses $(\mathbf{y}_1, \mathbf{y}_2)$ generated by the optimal policy π_r^* optimized under the reward $r(\mathbf{x}, \mathbf{y})$ and ranked by human experts or an oracle preference function $p^*(\cdot \mid \mathbf{y}_1, \mathbf{y}_2, \mathbf{x})$. Some methods, such as DPO, skip an explicit reward learning stage through a derivation to update the policy directly, which leads to implicit reward learning.
The policy optimization phase learns the LLM policy $\pi_r^*(\cdot \mid \mathbf{x})$ for a given reward $r(\mathbf{x}, \mathbf{y})$ by solving the KL-regularized policy optimization problem given as

$$\max_{\pi} \mathbb{E}_{\mathbf{x} \sim \mathcal{P}, \mathbf{y} \sim \pi(\cdot \mid \mathbf{x})} \Big[r(\mathbf{x}, \mathbf{y}) - \beta \mathbb{D}_{\text{KL}} \big[\pi(\cdot \mid \mathbf{x}) \parallel \pi_{\text{SFT}}(\cdot \mid \mathbf{x}) \big] \Big], \tag{2}$$

where $\beta > 0$ controls the deviation from the base reference policy π_{SFT}.

This process is repeated over multiple iterations as detailed in (Christiano et al., 2017; Lee et al., 2021; Park et al., 2022; Guo et al., 2024; Sharma et al., 2024; Lee et al., 2023) by performing oscillating updates to the policy and reward models until convergence.

Bilevel preference optimization formulation: We use the following bilevel formulation to represent the dependence of the policy-generated responses on the reward learning objective, which has also been showing in recent works by Chakraborty et al. (2023); Shen et al. (2024)

$$\begin{aligned} &\text{(upper)} \qquad \min_{r} \quad -\mathbb{E}_{[\mathbf{x} \sim \mathcal{P}, \mathbf{y}_i \sim \pi_r^*(\cdot \mid \mathbf{x}), (\mathbf{y}_w \succ \mathbf{y}_l) \sim p^*]} \big[\log \sigma (r(\mathbf{x}, \mathbf{y}_w) - r(\mathbf{x}, \mathbf{y}_l)) \big] \\ &\text{(lower)} \quad \text{s.t.} \ \pi_r^* := \arg\max_{\pi} \mathbb{E}_{\mathbf{x} \sim \mathcal{P}} \left[\mathbb{E}_{\mathbf{y} \sim \pi(\cdot \mid \mathbf{x})} \big[r(\mathbf{x}, \mathbf{y}) \big] - \beta \mathbb{D}_{\text{KL}} \big[\pi(\cdot \mid \mathbf{x}) \parallel \pi_{\text{SFT}}(\cdot \mid \mathbf{x}) \big] \right], \end{aligned} \tag{3}$$

where the upper level in eq. (3) represents the reward learning problem (refer to eq. (1)) and the lower level denotes the language model policy fine-tuning stage (refer to eq. (2)).

The bilevel optimization problem in eq. (3) is complex to solve. However, by utilizing the one-to-one equivalence between the reward function and the LLM policy (first shown in (Rafailov et al., 2023)), we can transform eq. (3) into an equivalent single-level form and solve it efficiently.

Due to the special structure of the equivalence between the reward function and the LLM policy, we can obtain the closed-form solution of the inner objective as

$$r(\mathbf{x}, \mathbf{y}) = \beta \log \frac{\pi_r^*(\mathbf{y} \mid \mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y} \mid \mathbf{x})} + \beta \log Z(\mathbf{x}). \tag{4}$$

Substituting this into eq. (3), we derive the new objective as

$$\max_{\pi^*(r)} J(\pi_r^*) = \mathbb{E}_{[\mathbf{x} \sim \mathcal{P}, \mathbf{y}_i \sim \pi_r^*(\cdot \mid \mathbf{x}), (\mathbf{y}_w \succ \mathbf{y}_l) \sim p^*]} \big[\log \sigma (\beta \log \frac{\pi_r^*(\mathbf{y}_w \mid \mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y}_w \mid \mathbf{x})} - \beta \log \frac{\pi_r^*(\mathbf{y}_l \mid \mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y}_l \mid \mathbf{x})}) \big], \tag{5}$$

where we replace the closed-form relationship between (π_r^*, r) from eq. (4) in eq. (3) to obtain eq. (5). Similar to (Rafailov et al., 2023), the above problem becomes an optimization in the space of π_r^*, which we solve via parametrization as

$$\max_{\theta} J(\theta) = \mathbb{E}_{[\mathbf{x} \sim \mathcal{P}, \mathbf{y}_i \sim \pi_\theta(\cdot \mid \mathbf{x}), (\mathbf{y}_w \succ \mathbf{y}_l) \sim p^*]} \big[\log \sigma (\beta \log \frac{\pi_\theta(\mathbf{y}_w \mid \mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y}_w \mid \mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_l \mid \mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y}_l \mid \mathbf{x})}) \big]. \tag{6}$$

We arrive at eq. (6) by parameterizing the policy, π_θ. By utilizing the closed-form relation eq. (4), the complexity of estimating the hyper-gradient is eliminated, and optimization becomes efficient. However, the policy parameter remains dependent on the trajectory distribution, akin to the policy gradient in reinforcement learning.

Gradient evaluation:

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta \sum_{\mathbf{x},\mathbf{y}_w,\mathbf{y}_l} \pi_\theta(\mathbf{y}_w \mid \mathbf{x})\pi_\theta(\mathbf{y}_l \mid \mathbf{x})\big[\log\sigma(\beta\log\frac{\pi_\theta(\mathbf{y}_w \mid \mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y}_w \mid \mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_l \mid \mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y}_l \mid \mathbf{x})})\big] \\ &= \nabla_\theta \sum_{\mathbf{x},\mathbf{y}_w,\mathbf{y}_l} \hat{\pi}_\theta(\mathbf{y}_w, \mathbf{y}_l \mid \mathbf{x})\big[F_\theta(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\big].\end{aligned} \tag{7}$$

Let $F_\theta(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l) = \log\sigma(\beta\log\frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y}_w|\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_l|\mathbf{x})}{\pi_{\text{SFT}}(\mathbf{y}_l|\mathbf{x})})$ and represent the distribution $\hat{\pi}_\theta(\mathbf{y}_w,\mathbf{y}_l \mid \mathbf{x}) = \pi_\theta(\mathbf{y}_w \mid \mathbf{x})\pi_\theta(\mathbf{y}_l \mid \mathbf{x})$.

With the above simplification, we can express the gradient as the sum of two gradient terms

$$\nabla_\theta J(\theta) = \underbrace{\sum_{\mathbf{x},\mathbf{y}_w,\mathbf{y}_l} \nabla_\theta\hat{\pi}_\theta(\mathbf{y}_w,\mathbf{y}_l \mid \mathbf{x})\big[F_\theta(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\big]}_{T_1} + \underbrace{\mathbb{E}_{[\mathbf{x}\sim\mathcal{P},\mathbf{y}_i\sim\pi_r^*(\cdot|\mathbf{x}),(\mathbf{y}_w\succ\mathbf{y}_l)\sim p^*]}\nabla_\theta F_\theta(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)}_{T_2}. \tag{8}$$

Note: The second term of the gradient expression in eq. (8), T_2, is the equivalent to the gradient found in Direct Preference Optimization frameworks (Rafailov et al., 2023). Our gradient expression differs because of a new term, T_1, which we simplify as

$$T_1 = \mathbb{E}\Big[\big(\nabla_\theta\log\pi_\theta(\mathbf{y}_w \mid \mathbf{x}) + \nabla_\theta\log\pi_\theta(\mathbf{y}_l \mid \mathbf{x})\big)F_\theta(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\Big]. \tag{9}$$

As shown in eq. (9), the gradient drives the generation of $\mathbf{y}_w$ and $\mathbf{y}_l$ to maximize the implicit reward function $F_\theta(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)$. This occurs when $\mathbf{y}_w$ and $\mathbf{y}_l$ are highly diverse, which means efficient exploration during sampling.

3 Experiments

The following experiments are my own individual contribution to the experimental evaluation of SAIL.

The experiment section aims to answer two major research questions: **RQ1**: *How does SAIL address the interdependence of model and data?* and **RQ2**: *Can SAIL be applied to practical, state-of-the-art LLM alignment?*

We test the SAIL formulation with different sentiment generation and question answering tasks using a pretrained GPT-2 model (Radford et al., 2019). For each setup, there is an additional hyperparameter, which is the *coefficient of added gradient* (i.e., the magnitude by which it deviates from the original DPO objective). The gradient in eq. (8) can be rewritten in the following format:

$$\nabla_\theta J(\theta) = \lambda T_1 + T_2 \tag{10}$$

Baselines. We primarily compare our method against standard Direct Preference Optimization (DPO) (Rafailov et al., 2023), as it represents a foundational offline alignment approach that balances both performance and efficiency. Proximal Policy Optimization (PPO) (Schulman et al., 2017) and other methods that involve full RL training require extensive computational resources and longer training times, making them less practical for large-scale online alignment tasks. Therefore, we do not focus on them as main baselines.

3.1 SAIL for Sentiment Generation

Figure 1 illustrates the experimental results derived from a modified version of the IMDB movie review dataset, initially used in the DPO paper (Rafailov et al., 2023). The objective of this task was to train a model to generate movie reviews with positive sentiment using a preference dataset and a reward oracle. **Reward oracle:** The reward oracle employed was a DistilBERT model fine-tuned for sentiment analysis (Sanh, 2019). **Dataset:** Our initial experiments revealed that the model could easily learn the original dataset with minimal training samples and iterations. To increase task complexity, we modified the dataset by first fine-tuning a model with DPO to generate negative samples. After fine-tuning, we used the negative-tuned model to produce several responses, selecting the two with the

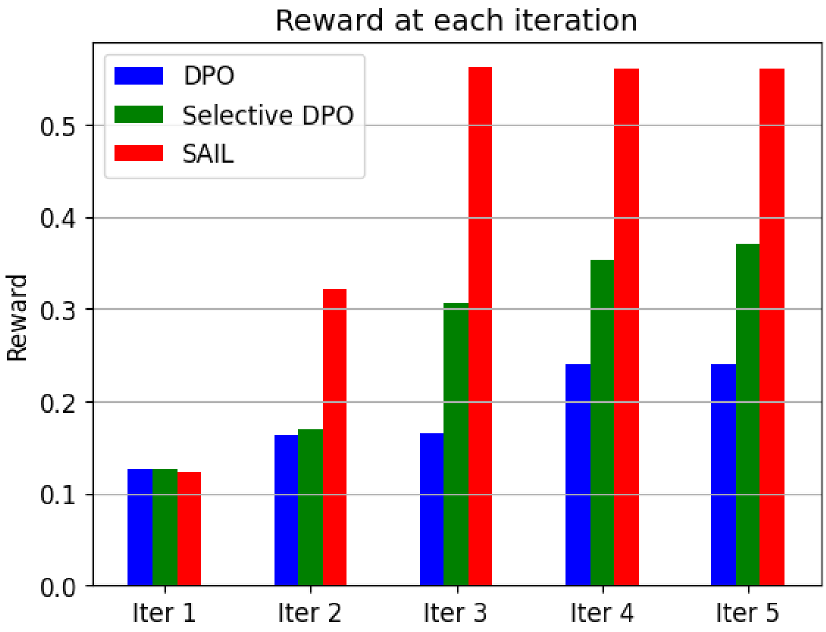

Figure 1: Normalized average sentiment score as reward per iteration on a 0 to 1 scale for the task of positive sentiment generation on the IMDB dataset. We found that a λ of approximately 0.3 was effective for sentiment generation tasks

highest and lowest sentiment scores to form the chosen and rejected responses. Using this modified dataset, we trained a different pretrained GPT-2 model with the SAIL method, as shown in fig. 1.

In each iteration, the model was trained on a preference dataset generated from the model's outputs in the previous iteration. Selective DPO, an approximation of SAIL, induced exploration by generating ten responses per prompt, in contrast to DPO and SAIL, which generated two responses. From the generated responses, the highest and lowest reward generations were selected to form the preference pair.

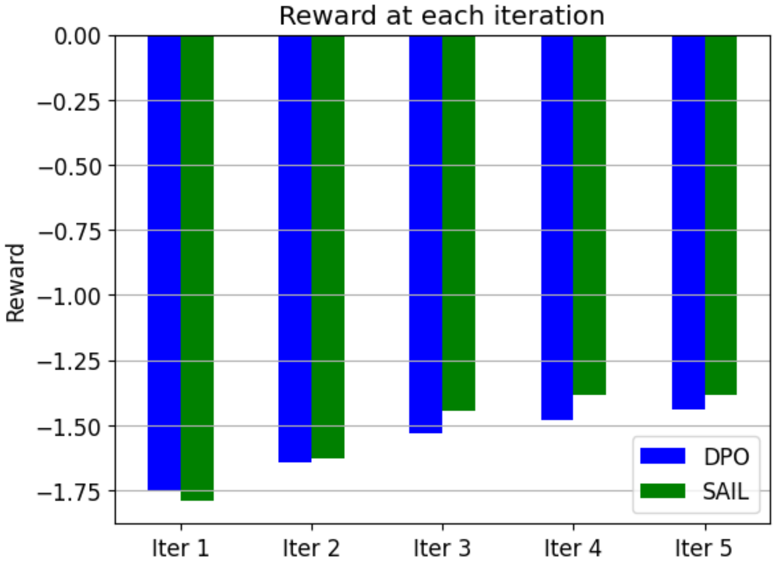

Figure 2: Unnormalized average sentiment score as reward per iteration for the task of generating non-toxic responses.

Figure 2 presents results for the same task, positive sentiment generation, applied to the Allen AI Real Toxicity Prompts dataset (Gehman et al., 2020). In this case, the reward oracle was a RoBERTa model trained to score text based on toxicity (Vidgen et al., 2020). Similar to the IMDB task, we increased the task difficulty by employing the same methodology to modify the dataset.

Technical details. The added gradient term in eq. (8) can be easily implemented and integrated into existing DPO pipelines[1] as they are complete gradients of the policy log-likelihood.

3.2 SAIL for Question Answering

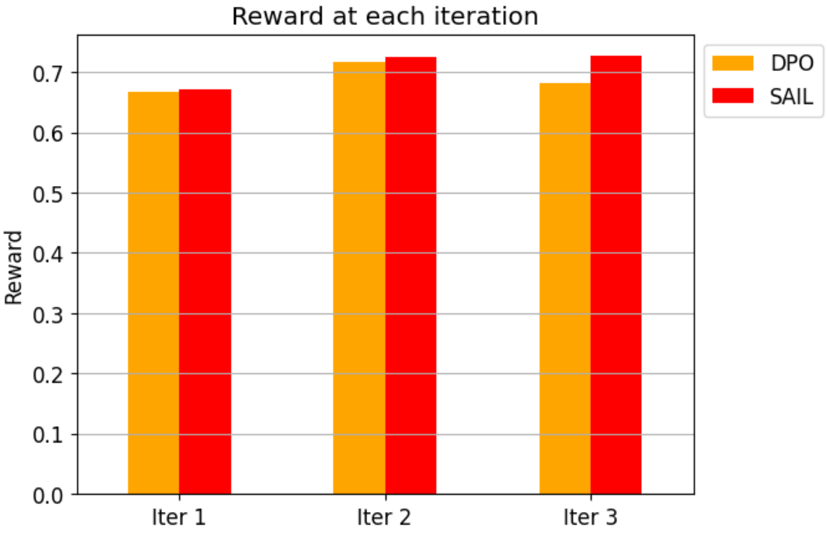

Figure 3: Average F1 score as reward per iteration for the task of question answering. SAIL demonstrates consistent improvements in performance from iteration to iteration and outperforms DPO. We found that a λ of approximately 0.1 was effective.

Figure 3 shows the performance of SAIL as compared to DPO for the task of question answering, demonstrating the transferability of SAIL to different domains. **Reward oracle:** In contrast to sentiment generation tasks, the reward oracle for this task was not a separate pretrained model, but instead the F1 score of a response. **Dataset:** We used the SQuAD 2.0 dataset (Rajpurkar et al., 2018).

4 Conclusions

This study reveals that online LLM alignment is rooted in a bilevel formulation, which can be simplified into an efficient single-level first-order approach. The SAIL method consistently outperforms DPO in sentiment generation and question-answering tasks, which demonstrates its effectiveness and adaptability in aligning large language models for various tasks.

[1] For example, my implementation is based on the popular and efficient DPOTrainer in TRL package `https://huggingface.co/docs/trl/main/en/dpo_trainer`.

References

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pp. 64–66. PMLR, 2020. 1

Souradip Chakraborty, Amrit Singh Bedi, Alec Koppel, Dinesh Manocha, Huazheng Wang, Mengdi Wang, and Furong Huang. Parl: A unified framework for policy alignment in reinforcement learning, 2023. 1, 2

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. Maxmin-rlhf: Towards equitable alignment of large language models with diverse human preferences, 2024. 1

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. 2

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020. 5

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. Direct language model alignment from online ai feedback, 2024. 1, 2

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. Rlaif: Scaling reinforcement learning from human feedback with ai feedback, 2023. 1, 2

Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training, 2021. 2

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 1

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning, 2022. 2

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL `https://api.semanticscholar.org/CorpusID:160025533`. 3

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023. 1, 2, 3

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, 2018. 5

V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 3

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

Archit Sharma, Sedrick Keh, Eric Mitchell, Chelsea Finn, Kushal Arora, and Thomas Kollar. A critical evaluation of ai feedback for aligning large language models, 2024. 1, 2

Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and rlhf, 2024. 1, 2

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback, 2024. 1

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*, 2020. 5

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models, 2024. 1