Align-Pro: A Principled Approach to Prompt Optimization for LLM Alignment

Prashant Trivedi[1], **Souradip Chakraborty**[2], **Avinash Reddy**[1], **Vaneet Aggarwal**[3], **Amrit Singh Bedi**[1], **George K. Atia**[1]

[1]University of Central Florida
[2]University of Maryland
[3]Purdue University
prashant.trivedi@ucf.edu, schakra3@umd.edu, av131693@ucf.edu, vaneet@purdue.edu, amritbedi@ucf.edu, george.atia@ucf.edu

Abstract

The alignment of large language models (LLMs) with human values is critical as these models become increasingly integrated into various societal and decision-making processes. Traditional methods, such as reinforcement learning from human feedback (RLHF), achieve alignment by fine-tuning model parameters, but these approaches are often computationally expensive and impractical when models are frozen or inaccessible for parameter modification. In contrast, prompt optimization is a viable alternative to RLHF for LLM alignment. While the existing literature has shown empirical promise of prompt optimization, its theoretical underpinning remains under-explored. We address this gap by formulating prompt optimization as an optimization problem and try to provide theoretical insights into the optimality of such a framework. To analyze the performance of the prompt optimization, we study theoretical suboptimality bounds and provide insights in terms of how prompt optimization depends upon the given prompter and target model. We also provide empirical validation through experiments on various datasets, demonstrating that prompt optimization can effectively align LLMs, even when parameter fine-tuning is not feasible.

Introduction

The quest to align large language models (LLMs) with human values is not just an academic pursuit but a practical necessity (Wang et al. 2024; Kaufmann et al. 2023). As these AI models (e.g., ChatGPT, Llamma2, etc.) increasingly become an essential part of various aspects of daily life and decision-making processes, ensuring their outputs reflect ethical considerations and societal norms becomes crucial (Li, Krishna, and Lakkaraju 2024; Dai et al. 2023). The standard approach to aligning LLMs has been through fine-tuning parameters via reinforcement learning from human feedback (RLHF) (Zhu, Jiao, and Jordan 2023; Azar et al. 2023; Ziegler et al. 2019a), which involves three main steps: Supervised Fine-Tuning (SFT), reward learning, and RL fine-tuning. However, this process can be resource-intensive, as it necessitates updating model parameters (Casper et al. 2023; Ouyang et al. 2022). A further complication to alignment arises when models are either 'frozen' or operate as 'black box,' where direct access to tweak parameters is restricted (Diao et al. 2023; Shin et al. 2020). These scenarios pose a critical question: How can we ensure LLM alignment when parameter updates are not allowed or possible?

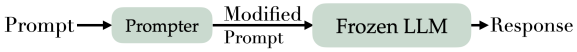


Figure 1: A basic overview of the prompt optimization framework. A prompter modifies the prompt before passing it through the target frozen LLM.

One promising solution lies in the concept of *prompt optimization* (Lin et al. 2024; Li and Liang 2018; Lester, Al-Rfou, and Constant 2021). This technique leverages the idea that the output of an LLM is a function of the input prompt—thereby turning the prompt into a powerful tool to elicit desired responses to align with specific rewards (cf. Figure 1). Various empirical studies in the literature have shown the significant benefits of prompt optimization techniques for LLM alignment (Shin et al. 2020; Kong et al. 2024; Wang et al. 2023b). However, theoretical insights about the working of prompt optimization have not been well studied. This raises an important question about the optimality of prompt optimization compared to traditional fine-tuning: *Can prompt optimization for LLM alignment achieve performance comparable to fine-tuning?*

In this work, we try to investigate and answer the above question. To the best of our knowledge, there is a notable absence of literature focusing on a theoretical formulation of prompt optimization specifically for LLM alignment. This paper aims to fill this gap by developing a unified optimization framework (called Align-Pro) to analyze prompt optimization for LLM alignment. We explore its theoretical performance, particularly in terms of suboptimality bounds, which measure how close the responses generated via the prompt optimization are to the outcomes obtained through fine-tuned models. Furthermore, we provide proof of concept empirical evidence to support the theoretical insights. We summarize our main contributions as follows.

- **An optimization framework to prompt optimization for LLM alignment.** We propose Align-Pro: a prompt optimization framework where we motivate the optimization objective, which would help reduce the suboptimality gap in the alignment. The optimization problem considered allows us to theoretically study the prompt optimization for LLM alignment. Following the standard analysis of LLM alignment, we derive a closed-form expression for the optimal prompt distribution.
- **We study the suboptimality of prompt optimization with respect to the fine-tuning method.** We establish theoretical bounds on the difference between the expected rewards obtained from the fine-tuned policy, which represents the benchmark for model performance, and the optimal policy derived from our prompt optimization approach.
- **Experimental results.** We conduct a series of experiments on three datasets to support the insights we obtain from the theoretical analysis. Align-Pro demonstrates better performance in terms of the mean rewards and win rate over the baseline without fine-tuning, showcasing its effectiveness across three datasets and diverse model configurations.

Related Work

RLHF and LLM fine-tuning: RLHF has become the most widely used method for aligning LLM responses with human values (Dubois et al. 2024; Ouyang et al. 2022; Ziegler et al. 2019b). For a more comprehensive discussion on RLHF, refer to some recent surveys (Casper et al. 2023; Chaudhari et al. 2024). Recently, some methods have been developed to bypass the need for RL, directly utilizing a preference dataset for alignment, including direct preference optimization (DPO) (Rafailov et al. 2024), SLiC (Zhao et al. 2023), and other extensions (Amini, Vieira, and Cotterell 2024; Azar et al. 2024; Gou and Nguyen 2024; Liu et al. 2024; Morimura et al. 2024; Tang et al. 2024; Wang et al. 2023a). The recent work of (Dwaracherla et al. 2024) has demonstrated the potential of efficient exploration methods to improve LLM responses based on human preference feedback. Moreover, methods such as ORPO (Hong, Lee, and Thorne 2024) align the model without using a reference model. Furthermore, intuitive fine-tuning (IFT) conducts alignment solely relying on positive samples and a single policy, starting from a pre-trained base model (Hua et al. 2024). However, all of these approaches are focused on alignment via parameter fine-tuning.

Prompt optimization for alignment: Prompt optimization has seen significant growth in recent years. Early efforts focused on white-box LLMs, such as AutoPrompt (Shin et al. 2020) and FluentPrompt (Shi et al. 2023), which used gradient-based methods to generate prompts from labeled data. Soft prompt methods, such as (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021; Zhong, Friedman, and Chen 2021), also gained traction. Recently, the focus has shifted to optimizing prompts for black-box LLMs. Techniques like clip-tuning (Chai et al. 2022), BBT (Sun et al. 2022b), and BBTv2 (Sun et al. 2022a) optimize prompts by leveraging input embeddings and output logits, often using low-dimensional subspace optimization. Some approaches use RL ideas for prompt optimization for alignment, including BDPL (Diao et al. 2023), PRewrite (Kong et al. 2024), and MultiPrompter (Kim et al. 2023), which iteratively update prompts. Planning-based approaches, such as PromptAgent (Wang et al. 2023b), have also gained attention. Additionally, APOHF (Lin et al. 2024) leverages dueling bandits theory to refine prompts using preference feedback. However, theoretical connections in terms of comparing the performance of prompt optimization with the fine-tuning approach are not studied in detail.

Other works with similar formulations: Beyond prompt optimization and fine-tuning, other areas share similar theoretical formulations. For instance, (Hong et al. 2024; Perez et al. 2022; Wichers, Denison, and Beirami 2024; Lee et al. 2024; Beetham et al. 2024) explore automated red teaming by training a red team LLM with reinforcement learning to generate test cases that provoke undesirable responses from a target LLM. While the context differs, the red team model's training objective aligns closely with our prompt optimization objective. In contrast, in this work, we motivate the selection of objectives for prompt optimization and focus on understanding the suboptimality of prompt optimization with respect to fine-tuned models.

Preliminaries and Background

This section provides the essential background and foundational concepts relevant to alignment. We start by defining the notation, followed by a quick overview of the RLHF framework, which involves three key steps: (i) supervised fine-tuning (SFT), (ii) reward learning, and (iii) fine-tuning with RL.

Language Models. We start by defining the language model mathematically. Let us denote the vocabulary set by $\mathcal{V}$, and we denote the language model by $\pi(y|x)$, which takes in the sequence of tokens $x := \{x_1, x_2, \cdots, x_N\}$ (with each $x_i \in \mathcal{V}$) as an input, and generate response $y := \{y_1, y_2, \cdots, y_M\}$ (with each $y_i \in \mathcal{V}$) as the output. At instant t, each output token $y_t \sim \pi(\cdot|x_t)$.

Supervised Fine-Tuning (SFT). SFT is the initial step in the RLHF process. It involves fine-tuning a pre-trained LLM on a vast dataset of human-generated text in a supervised manner.

Reward Learning. This stage involves learning the reward model by gathering preferences from experts/human feedback or an oracle based on outputs generated by the SFT model denoted by π_{sft}. The optimization is generally performed under the Bradley-Terry model for pairwise comparison (Bradley and Terry 1952), which seeks to minimize the loss formulated as:

$$\mathcal{L}(r, D_r) = -\mathbb{E}_{(x,y_u,y_v)\sim D_r}\left[\log\left(\sigma(r(x, y_u) - r(x, y_v))\right)\right] \tag{1}$$

where D_r denotes the dataset of response pairs (y_u, y_v), with y_u and y_v representing the winning and the losing

responses, respectively, which are generated by the policy π_{sft} optimized under the reward $r(x, y)$, and evaluated by human experts or an oracle function $p^*(\cdot|y_u, y_v, x)$, and $\sigma(\cdot)$ is the sigmoid function.

Fine-tuning with RL. In this step, we obtain the aligned model which maximizes the reward model $r(x, y)$ (trained in the previous step) by solving a KL-regularized optimization problem:

$$\max_{\pi} \mathbb{E}_{x\sim P, y\sim\pi(\cdot|x)} \left[r(x, y) - \beta\mathbb{D}_{KL}(\pi(\cdot|x)\|\pi_{\text{sft}}(\cdot|x))\right], \tag{2}$$

where, $\beta > 0$ is a parameter that controls the deviation from the baseline policy π_{sft}. This iterative process alternates between updating the policy and reward models until convergence, as detailed in previous works (Kaufmann et al. 2023; Zhu, Jiao, and Jordan 2023).

Prompt Optimization Framework for LLM Alignment

In this section, we provide a mathematical formulation for the framework of prompt optimization for LLM alignment. In traditional LLM alignment, as described in (2), the model parameters are fine-tuned to adjust the response distributions in a way that maximizes the reward function. However, in our setting, we operate under a different regime, starting with a pre-trained language model, denoted by π_F, whose parameters remain frozen. In this case, direct modification of the model to align with a reward function is not allowed. Therefore, an alternative and widely adopted approach in the literature is to optimize the input prompt itself to yield better-aligned responses (Kong et al. 2024; Shin et al. 2020; Zhou et al. 2023). Typically, this process involves iterative prompt refinement, where the model outputs are evaluated and compared to human preferences, and the prompts are adjusted accordingly. However, such iterative fine-tuning can be computationally expensive and time-intensive.

Interestingly, although we cannot fine-tune the frozen model π_F, we can fine-tune the prompter model ρ in any desired manner. However, a fundamental challenge arises: what should be the objective for optimizing the prompter? While substantial empirical evidence in the literature demonstrates that prompt optimization can significantly enhance response generation and improve alignment (Shin et al. 2020; Kong et al. 2024; Zhou et al. 2023), there is no specific emphasis on developing a mathematical framework to guide this process. We start by addressing this gap as follows.

Optimization Objective for Prompter Design. First, we revisit the basics of LLM alignment. For a given prompt x, the probability of generating a response y from the frozen model is represented by $\pi_F(y|x)$. After introducing the prompter model ρ, the probability of generating response y given input x (denoted by $\widetilde{\pi}_\rho$) can be expressed as:

$$\widetilde{\pi}_\rho(y|x) = \sum_{x'} \pi_F(y|x')\rho(x'|x), \tag{3}$$

which captures the probability of generating the response y for a given x under the influence of the prompter ρ. Let us consider the ideal scenario: if we were able to fine-tune the language model π_F, we would solve the optimization problem in (2) and obtain the RLHF optimal solution π^*, which is given by (Peng et al. 2019; Peters and Schaal 2007)

$$\pi^*(y|x) = \frac{1}{Z^*(x)}\pi_F(y|x)\exp\left(\frac{r^*(x, y)}{\beta}\right), \tag{4}$$

where $Z^*(x) = \sum_y \pi_F(y|x)\exp(r^*(x, y)/\beta)$ is the normalizing constant, and β is the alignment tuning parameter, and reward r^* is obtained from solving (1). We emphasize that if we have a prompter ρ that performs as well as the RLHF-optimal policy π^*, it should be a sufficient indicator of a good prompter. With this understanding, we consider the following prompter suboptimality gap given by

$$\triangle(\rho) := J(\pi^*) - J(\widetilde{\pi}_\rho), \tag{5}$$

which captures how well our prompter is doing with respect to fine-tuned optimal policy π^*. Mathematically, it holds that

$$\begin{aligned} &J(\pi^*) - J(\widetilde{\pi}_\rho) \\ &= \mathbb{E}_{x\sim P, y\sim\pi^*(\cdot|x)}[r^*(x, y)] - \mathbb{E}_{x\sim P, y\sim\widetilde{\pi}_\rho(\cdot|x)}[r^*(x, y)] \\ &= \mathbb{E}_{x\sim P}\left[\mathbb{E}_{y\sim\pi^*(\cdot|x)}[r^*(x, y)] - \mathbb{E}_{\substack{x'\sim\rho(\cdot|x) \\ y\sim\pi_F(\cdot|x')}}[r^*(x, y)]\right]. \end{aligned} \tag{6}$$

Equation (6) evaluates the difference in expected return between the optimal RLHF policy π^* and our prompt optimization policy $\widetilde{\pi}_\rho$, indicating how much better (or worse) π^* performs compared to $\widetilde{\pi}_\rho$. We highlight that this performance gap is clearly influenced by the choice of the prompt distribution ρ; a non-optimal ρ can result in a significant gap. This leads us to the following questions:

- **Q1**: Can we design an optimal prompter ρ^* that closes the suboptimality gap between the fine-tuned policy π^*, and the prompt optimization policy $\widetilde{\pi}_{\rho^*}$ as mentioned in Equation (6)?
- **Q2**: If such a ρ^* exists, then can $\widetilde{\pi}_{\rho^*}$ outperform the fine-tuned optimal policy π^*?

We address these questions in the next section.

Proposed Approach: Align-Pro

Let us start by addressing Q1 and develop a general prompt optimization framework to design an optimal prompter ρ^*. But then the first question arises: in what sense is ρ^* optimal? In order to see that, let us reconsider $J(\pi^*) - J(\widetilde{\pi}_\rho)$ and after adding-subtracting $\mathbb{E}_{y\sim\pi_F(\cdot|x)}[r^*(x, y)]$ in the right hand side of Equation (6), we get

$$J(\pi^*) - J(\widetilde{\pi}_\rho) = \mathbb{E}_{x\sim P}[\Delta_1 + \Delta_2], \tag{7}$$

where Δ_1 and Δ_2 are defined as

$$\begin{aligned} \Delta_1 &:= \mathbb{E}_{y\sim\pi^*(\cdot|x)}[r^*(x, y)] - \mathbb{E}_{y\sim\pi_F(\cdot|x)}[r^*(x, y)] \\ \Delta_2 &:= \mathbb{E}_{y\sim\pi_F(\cdot|x)}[r^*(x, y)] - \mathbb{E}_{y\sim\widetilde{\pi}_\rho(\cdot|x)}[r^*(x, y)] \\ &= \mathbb{E}_{y\sim\pi_F(\cdot|x)}[r^*(x, y)] - \mathbb{E}_{\substack{x'\sim\rho(\cdot|x) \\ y\sim\pi_F(\cdot|x')}}[r^*(x, y)]. \end{aligned}$$

We remark that in (7), Δ_1 is the suboptimality gap between the optimal fine-tuned policy, and the frozen model π_F. Thus, it captures the effectiveness of the optimal RLHF policy with respect to the frozen model. In other words, it quantifies how good or bad our frozen model is with respect to the optimally aligned model. We note that Δ_1 is constant for a given π_F and does not depend upon prompter ρ, hence we cannot improve this part with the prompter. Another insight is that since π^* is the optimal RLHF policy, $\Delta_1 \geq 0$, i.e., is always positive. On the other hand, the second term, Δ_2, depends upon our prompter ρ and can be controlled by designing a prompter. This observation leads to the formulation of an optimization problem for the prompter as follows.

Optimization Problem for Prompter

We recall from the definition of Δ_2 that we would need to learn a ρ such that Δ_2 is minimized. To achieve that, we recognize that the only term involving the prompter ρ in Δ_2 is $\mathbb{E}_{x' \sim \rho(\cdot|x), y \sim \pi_F(\cdot|x')}[r^*(x,y)]$, and minimizing Δ_2, we need to solve the following optimization problem

$$\max_{\rho} \mathbb{E}_{x' \sim \rho(\cdot|x), y \sim \pi_F(\cdot|x')}[r^*(x,y)]. \tag{8}$$

However, at the same time, since our prompter is also another language model, we will already have access to a baseline supervised fine-tuned prompter ρ_{sft}, and we want to ensure that our prompter ρ^* does not deviate significantly from ρ_{sft}, which motivates us to include a known and supervised fine-tuned prompter, denoted by ρ_{sft}. Thus, we solve the following optimization problem:

$$\max_{\rho} \mathbb{E}_{x \sim P} \mathbb{E}_{\substack{x' \sim \rho(\cdot|x) \\ y \sim \pi_F(\cdot|x')}} [r^*(x,y)] - \lambda \mathbb{D}_{KL}(\rho(\cdot|x) \| \rho_{\text{sft}}(\cdot|x)). \tag{9}$$

We have introduced a KL-divergence-based regularizer above between the prompter ρ and a reference supervised fine-tuned prompter ρ_{sft}. This helps with the development of a proper optimization problem with a closed-form expression and enables control over proximity to the initial prompter ρ_{sft} through the tuning parameter λ. We note that the formulation in (9) has also appeared in the red teaming literature for learning an attacker promoter (Hong et al. 2024; Perez et al. 2022; Wichers, Denison, and Beirami 2024; Lee et al. 2024; Beetham et al. 2024).

Interpretation of λ. Another interesting interpretation of λ is that it controls the extent of prompt optimization we want to introduce into the pipeline, hence we also refer to it as the prompt tuning parameter. For instance, $\lambda \to \infty$ means no prompt optimization, while $\lambda \to 0$, drives the optimization toward maximizing the prompter reward, albeit at the cost of deviating from ρ_{sft} which might be important in certain cases. Therefore, λ provides a meaningful trade-off, and its effects will be further elucidated in the following section.

The following Lemma 1 provides the optimal solution to the optimization problem (9).

Lemma 1. *Let $R(x,x') := \mathbb{E}_{y \sim \pi_F(\cdot|x')}[r^*(x,y)]$, and $\lambda > 0$ be the prompter tuning parameter. The optimal prompt distribution ρ^* that maximizes the objective function of the optimization problem* (9) *is given by:*

$$\rho^*(x'|x) = \frac{1}{Z(x)} \rho_{\text{sft}}(x'|x) \exp\left(\frac{1}{\lambda} R(x,x')\right), \tag{10}$$

where $Z(x)$ is the log partition function given by

$$Z(x) = \sum_{x'} \rho_{\text{sft}}(x'|x) \exp\left(\frac{1}{\lambda} R(x,x')\right).$$

The proof is available in Appendix A (Trivedi et al. 2025) and follows from the derivations in Rafailov et al. (2023). Next, we move on to answer Q2, in which we utilize the optimal prompter $\rho^*(x'|x)$ to obtain a bound on the suboptimality gap. Notably, the integration of this optimal prompter with the frozen model will lead to the refined performance expressed in terms of the modified optimal policy $\widetilde{\pi}^*_\rho(y|x) = \sum_{x'} \rho^*(x'|x)\pi_F(y|x')$. This will capture the effectiveness of the prompt optimization process and offer insights into how closely the modified policy $\widetilde{\pi}_{\rho^*}$ approximates the true optimal policy π^*.

Theoretical Insights w.r.t Fine-Tuning

We begin by establishing a bound on the suboptimality gap for the optimal prompter. The following theorem bounds the suboptimality gap $J(\pi^*) - J(\widetilde{\pi}_{\rho^*})$ when the optimal prompter ρ^* as obtained in Lemma 1 is used. We present our result in Theorem 1 as follows. The detailed proof is available in Appendix B (Trivedi et al. 2025).

Theorem 1. *Let the optimal prompter $\rho^*(x'|x)$ be given as in Equation* (10)*. Then, the suboptimality gap is bounded as*

$$\begin{aligned} J(\pi^*) - J(\widetilde{\pi}_{\rho^*}) \leq\ & r_{\max} \mathbb{E}_{x \sim P}[d_{TV}(\pi^*(\cdot|x), \pi_F(\cdot|x))] \\ & + r_{\max} \mathbb{E}_{x \sim P, x' \sim \rho_{\text{sft}}(\cdot|x)}[d_{TV}(\pi_F(\cdot|x), \pi_F(\cdot|x'))] \\ & - \lambda\, \mathbb{E}_{x \sim P}[\mathbb{D}_{KL}(\rho^*(\cdot|x) \| \rho_{\text{sft}}(\cdot|x))], \end{aligned} \tag{11}$$

where P denotes the prompt distribution, λ is the prompter tuning parameter, and d_{TV} is the total variation distance.

Theorem 1 provides an upper bound on the suboptimality gap between an optimal RLHF policy π^* and the optimal policy obtained by the prompt optimization approach $\widetilde{\pi}_{\rho^*}$. We now provide the interpretations to each term of the suboptimality gap given in Theorem 1.

- **Significance of first term in RHS of** (11)**:** The first term in Equation (11) is always non-negative. It captures the intrinsic difficulty of obtaining the optimal RLHF policy via a prompt optimization setup when the frozen model is not fully aligned. We note that when $\pi_F = \pi^*$, the first term in Theorem 1 becomes zero. However, this scenario is not relevant to our prompt optimization framework, as it necessitates fine-tuning the frozen LLM.
- **Significance of second term in RHS of** (11)**:** This term measures how much the response distribution the frozen policy π_F changes when its input changes from x to x' under ρ_{sft}. For ρ_{sft} as delta distribution, this term will be zero, which essentially implies that this term is trying to capture the variation in the prompts (which should be minimal) due to the introduction of ρ_{sft} into the formulation.

- **Significance of third term in RHS of** (11)**:** The third term captures the KL divergence between the optimal prompter ρ^* and the given prompter ρ_{sft}. This term is important because it explains that we can reduce the suboptimality bound via prompt optimization, which is making ρ^* far from ρ_{sft}, which can be controlled by the parameter λ.

Another interesting insight is that the upper bound on the suboptimality remains non-negative for $\mathbb{D}_{KL}(\rho^*(\cdot|x)\|\rho_{\text{sft}}(\cdot|x)) \leq \frac{\epsilon_1+\epsilon_2}{\lambda}$, where $\epsilon_1 := d_{TV}(\pi^*(\cdot|x), \pi_F(\cdot|x))$ and $\epsilon_2 := \mathbb{E}_{x'\sim\rho_{\text{sft}}(\cdot|x)}\left[d_{TV}(\pi_F(\cdot|x), \pi_F(\cdot|x'))\right]$. This essentially provide insight that in practice, with a budget of $\frac{\epsilon_1+\epsilon_2}{\lambda}$ for the prompter optimization can be sufficient to achieve performance similar to RLHF based fine tuning. This further highlights that we won't need to choose an optimal prompter arbitrarily far from the base prompt distribution, thereby preventing a significant loss in the quality (e.g., perplexity) of the generated outputs.

Experimental Evaluations

In this section, we present proof of concept experiments to validate the theoretical insights of our proposed prompt optimization framework, which we named Align-Pro. We outline our experimental setup, including the dataset, model architecture, and evaluation metrics. Following this, we present our results and provide a detailed analysis of our findings. Additional details on the experimental setup and evaluations can be found in Appendix C (Trivedi et al. 2025).

Experimental Setup

We evaluate the performance of our Align-Pro using two distinct prompter models, denoted as P1 (Phi-3.5-Instruct) and P2 (Qwen-2.5-1.5B-Instruct), which modifies and updates the original prompt. Additionally, we use two frozen models, denoted as F1 (Llama-3.1-8B-Instruct) and F2 (Llama-3.1-8B-Instruct) to generate the final responses. This setup results in four unique model architectures, each representing a combination of the prompter and frozen models. For each architecture, we assess performance for the following three different configurations.

- **No Fine-Tuning**: In this configuration, the prompter is not used, and only the frozen model is used to generate responses without any fine-tuning or prompt modifications.
- **Align-Pro**: In this setup, a fine-tuned prompter is placed before a frozen model. The prompter refines the input prompt, and the frozen model generates the response based on the optimized prompt.
- **RLHF**: In this configuration, the frozen model undergoes fine-tuning through RLHF, and the response is generated directly from this fine-tuned model.

Datasets: To capture the diversity in our experimental evaluations, we evaluate the performance over different datasets:

- **UltraFeedback** (Cui et al. 2024) : A large-scale, high-quality, and diversified AI feedback dataset which contains feedback from user-assistant conversations from various aspects. This dataset evaluates the coherence of the prompt-response pairs.
- **HelpSteer** (Wang et al. 2023c): A multi-attribute helpfulness dataset annotated for correctness, coherence, complexity, and verbosity in addition to overall helpfulness of responses.
- **Orca** (Mukherjee et al. 2023): This dataset features responses with detailed explanations for each prompt, promoting thinking and effective instruction-following capabilities in the models.

Evaluation Criteria. The primary objective of our experiments is to optimize the input prompt to guide the frozen LLM that produces the desired response effectively. We fine-tune the prompter using proximal policy optimization (PPO) within the RLHF framework to achieve this. The reward signal for this fine-tuning process is derived from the quality of the enhanced prompt and the output generated by the frozen LLM. We assess the performance of Align-Pro based on three key metrics: mean reward, variance, and win-rate comparison against the no-fine-tuning baseline.

Computational Resources. Since we do not alter the parameters of the frozen model, our experiments require relatively fewer computational resources. Consequently, we were able to conduct all our experiments using a machine equipped with an INTEL(R) XEON(R) GOLD 6526Y processor with a Nvidia H100 GPU. We used Python 3.11 to execute the experiments. we used the *PPOTrainer* variant from Hugging Face TRL library to run the RLHF and Prompt Optimization pipeline experiments.

Hyper-parameters. All of our experiments use the open-access TRL library, which is publicly available. The library can be accessed using the link[1]. For our experiments, we do not perform any extra hyper-parameter tuning; rather, we use the parameters *learning rate* $= 1.41e-5$ given in the above-mentioned link. Moreover, we use the following generation configurations to generate the response for evaluation in all experiments: temperature = 1.5, top P = 0.6 and top $K = 20$.

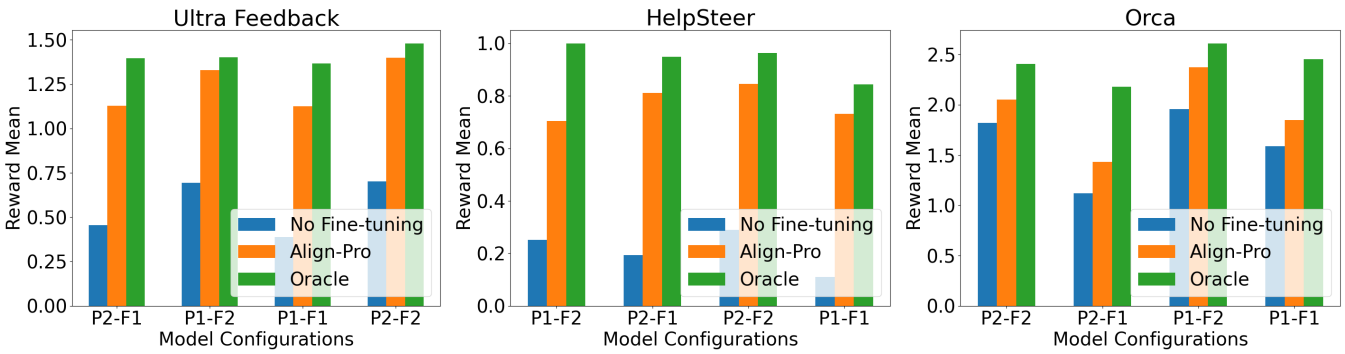
Results

Mean reward and variance comparison: We calculate mean rewards and variances to assess the quality of preferred response generation and the diversity of the language model for all configurations and different model architectures. To associate the reward to each response, we use the available reward model[2], which scores the response. This reward model is trained to assign higher scores to the responses that comply with the off-target attributes.

We also compared Align-Pro with an oracle model, where the LLM is fine-tuned using RLHF. Figure 2 presents the mean rewards across all three datasets for each model configuration, while Figure 3 shows the

[1]https://github.com/huggingface/trl/blob/main/examples/notebooks/gpt2-sentiment.ipynb

[2]https://huggingface.co/weqweasdas/RM-Gemma-2B



(a) (b) (c)

Figure 2: **Reward mean comparisons.** Figure shows the reward mean across the chosen datasets. Align-Pro shows an improvement over the no fine-tuning approach. We employ two prompters P1 (Phi-3.5-Instruct) and P2 (Qwen-2.5-1.5B-Instruct), along with two frozen LLMs, denoted as F1 (Llama-3.1-8B-Instruct) and F2 (Llama-3.1-8B-Instruct). The oracle is fine-tuned LLM via RLHF.

corresponding reward variances. Interestingly, Align-Pro consistently outperforms the baseline (no fine-tuning) in terms of mean reward, demonstrating its ability to generate more preferred and stable responses, leveraging prompt optimization and getting close to the performance of fine-tuned model denoted by oracle. Moreover, the variance in reward for Align-Pro is the lowest, indicating that it produces more reliable and stable outputs. In each figure, we employ two prompters, denoted as P1 (Phi-3.5-Instruct) and P2 (Qwen-2.5-1.5B-Instruct), along with two frozen LLMs, denoted as F1 (Llama-3.1-8B-Instruct) and F2 (Llama-3.1-8B-Instruct).

Win rate comparison: We evaluate the performance of our Align-Pro method by comparing it to the no fine-tuning configuration using win rate as the primary performance metric. We rely on GPT-4 as an external, impartial judge to ensure unbiased evaluation. The evaluation criteria focus on critical aspects of the response: helpfulness, harmlessness, relevance, accuracy, depth, creativity, and level of detail. To update the prompt, we use a standardized system prompt template. Table 1 presents the win rates for Align-Pro (denoted by A) against the no fine-tuning baseline (denoted by B). The results clearly show that, on average, Align-Pro significantly outperforms the no fine-tuning approach across all model architectures and datasets. These findings demonstrate the effectiveness of Align-Pro framework, which enhances performance by optimizing the input prompt while keeping the LLM frozen.

Summary: Our experiments confirm that using a prompter alongside a frozen LLM significantly enhances alignment. Moreover, the expected reward and the win-rate differences are affected by the degree to which the prompter and frozen model align with human preferences. These experimental results, therefore, support our theoretical insights. We include several examples using the full prompt rewriting, illustrating the original prompt, the re-written prompt, and the corresponding final response in the Appendix C (Trivedi et al. 2025).

Model Architectures Prompter, Frozen LLM	**UltraFeed (win rate)**		**HelpSteer (win rate)**		**Orca (win rate)**	
	A	**B**	**A**	**B**	**A**	**B**
Phi-3.5-Instruct, Llama-3.1-8B-Instruct	**60**	24	**46**	37	**63**	26
Qwen-2.5-1.5B-Instruct, Llama-3.1-8B-Instruct	**65**	23	**67**	23	**63**	30
Phi-3.5-Instruct, Qwen-2.5-7B-Instruct	**59**	27	**58**	27	**46**	**46**
Qwen-2.5-1.5B-Instruct, Qwen-2.5-7B-Instruct	**56**	30	**59**	25	**59**	27

Table 1: The table presents the win rates (for 100 samples) of our Align-Pro method, denoted by **A**, compared to the baseline no fine-tuning method, denoted by **B**. A higher win rate indicates superior performance. Bolded numbers highlight the higher win rates. Across all model architectures and datasets, Align-Pro consistently outperforms the no fine-tuning baseline, demonstrating its effectiveness in improving response quality.

Remark 1. *Our aim is not to present the best prompt optimizer but to develop an optimization framework for prompt optimization, which can help develop some theoretical insights into the performance of the prompt optimization approach. We seek to understand its theoretical performance relative to RLHF and fine-tuning methods, hence we did not compare our approach with other existing prompt optimization methods in the literature.*

Conclusion, Limitations and Future Work

This work introduces an optimization framework for prompt optimization by utilizing a smaller, trainable model to generate optimized prompts for a frozen large language model (LLM). This approach reduces computational costs while preserving the LLM's pre-trained capabilities. We provide a closed-form expression for the optimal prompter

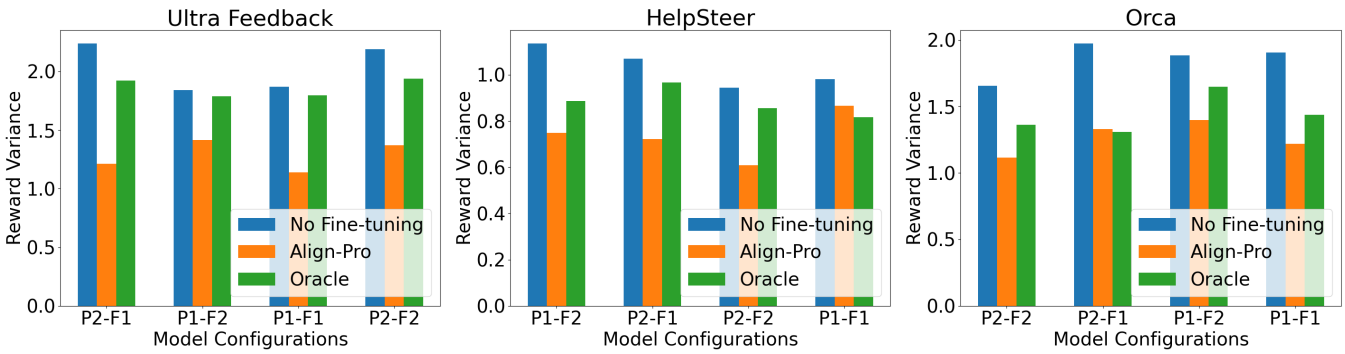


Figure 3: **Reward variance comparisons.** Align-Pro has the least variance compared to Oracle and no fine-tuning approach. Due to the prompter's precise guidance, the frozen LLM generates almost similar responses in terms of helpfulness and coherence, which results in less diverse responses. We use the following terminologies for the prompters and the frozen models: P1 (Phi-3.5-Instruct), P2 (Qwen-2.5-1.5B-Instruct), F1 (Llama-3.1-8B-Instruct), and F2 (Llama-3.1-8B-Instruct), respectively.

and use it to establish an upper bound on the suboptimality gap that compares the optimized prompt policy with the standard RLHF policy. We demonstrate the effectiveness of our method on three datasets and various model configurations. In each scenario, we observe that Align-Pro is better in terms of the mean rewards and win rate compared to the baseline with no fine-tuning.

Limitations and future work: Our framework is inherently limited by the capabilities of the frozen language model. Another limitation includes the sensitivity of the prompt to the final response; a slight change in the prompt can lead to profound changes in the final responses. Theoretically, it would also be interesting to develop lower bounds on suboptimality and to develop further insights into the performance of prompt optimization. We will consider some of these issues as part of our future work. Some other potential future directions of our work include analyzing the robustness of the optimal prompter in the presence of noise in the frozen model and exploring the use of multiple prompters in sequence before inputting them into the frozen model.

Acknowledgements

This work was supported in part by NSF under Award CCF-2106339 and DARPA under Agreement No. HR0011-24-9-0427.

References

Amini, A.; Vieira, T.; and Cotterell, R. 2024. Direct Preference Optimization with an Offset. *arXiv preprint arXiv:2402.10571*.

Azar, M. G.; Guo, Z. D.; Piot, B.; Munos, R.; Rowland, M.; Valko, M.; and Calandriello, D. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, 4447–4455. PMLR.

Azar, M. G.; Rowland, M.; Piot, B.; Guo, D.; Calandriello, D.; Valko, M.; and Munos, R. 2023. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*.

Beetham, J.; Chakraborty, S.; Wang, M.; Huang, F.; Bedi, A. S.; and Shah, M. 2024. LIAR: Leveraging Alignment (Best-of-N) to Jailbreak LLMs in Seconds. *arXiv preprint arXiv:2412.05232*.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Casper, S.; Davies, X.; Shi, C.; Gilbert, T. K.; Scheurer, J.; Rando, J.; Freedman, R.; Korbak, T.; Lindner, D.; Freire, P.; et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Chai, Y.; Wang, S.; Sun, Y.; Tian, H.; Wu, H.; and Wang, H. 2022. Clip-Tuning: Towards Derivative-free Prompt Learning with a Mixture of Rewards. In *Proc. EMNLP (Findings)*, 108–117.

Chaudhari, S.; Aggarwal, P.; Murahari, V.; Rajpurohit, T.; Kalyan, A.; Narasimhan, K.; Deshpande, A.; and da Silva, B. C. 2024. RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs. *arXiv preprint arXiv:2404.08555*.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; He, B.; Zhu, W.; Ni, Y.; Xie, G.; Xie, R.; Lin, Y.; et al. 2024. ULTRAFEEDBACK: Boosting Language Models with Scaled AI Feedback. In *Forty-first International Conference on Machine Learning*.

Dai, J.; Pan, X.; Sun, R.; Ji, J.; Xu, X.; Liu, M.; Wang, Y.; and Yang, Y. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*.

Diao, S.; Huang, Z.; Xu, R.; Li, X.; Yong, L.; Zhou, X.; and Zhang, T. 2023. Black-Box Prompt Learning for Pre-trained Language Models. *Transactions on Machine Learning Research*.

Dubois, Y.; Li, C. X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P. S.; and Hashimoto, T. B. 2024. Alpacafarm: A simulation framework for methods that learn

from human feedback. *Advances in Neural Information Processing Systems*, 36.

Dwaracherla, V.; Asghari, S. M.; Hao, B.; and Van Roy, B. 2024. Efficient Exploration for LLMs. *arXiv:2402.00396*.

Gou, Q.; and Nguyen, C.-T. 2024. Mixed Preference Optimization: Reinforcement Learning with Data Selection and Better Reference Model. *arXiv preprint arXiv:2403.19443*.

Hong, J.; Lee, N.; and Thorne, J. 2024. ORPO: Monolithic Preference Optimization without Reference Model. arXiv:2403.07691.

Hong, Z.-W.; Shenfeld, I.; Wang, T.-H.; Chuang, Y.-S.; Pareja, A.; Glass, J.; Srivastava, A.; and Agrawal, P. 2024. Curiosity-driven red-teaming for large language models. *arXiv preprint arXiv:2402.19464*.

Hua, E.; Qi, B.; Zhang, K.; Yu, Y.; Ding, N.; Lv, X.; Tian, K.; and Zhou, B. 2024. Intuitive Fine-Tuning: Towards Simplifying Alignment into a Single Process. arXiv:2405.11870.

Kaufmann, T.; Weng, P.; Bengs, V.; and Hüllermeier, E. 2023. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*.

Kim, D.-K.; Sohn, S.; Logeswaran, L.; Shim, D.; and Lee, H. 2023. MultiPrompter: Cooperative Prompt Optimization with Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2310.16730*.

Kong, W.; Hombaiah, S. A.; Zhang, M.; Mei, Q.; and Bendersky, M. 2024. PRewrite: Prompt Rewriting with Reinforcement Learning. *arXiv preprint arXiv:2401.08189*.

Lee, S.; Kim, M.; Cherif, L.; Dobre, D.; Lee, J.; Hwang, S. J.; Kawaguchi, K.; Gidel, G.; Bengio, Y.; Malkin, N.; et al. 2024. Learning diverse attacks on large language models for robust red-teaming and safety tuning. *arXiv preprint arXiv:2405.18540*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proc. EMNLP*, 3045–3059.

Li, A. J.; Krishna, S.; and Lakkaraju, H. 2024. More RLHF, More Trust? On The Impact of Human Preference Alignment On Language Model Trustworthiness. *arXiv preprint arXiv:2404.18870*.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proc. ACL*, 4582–4597.

Li, Y.; and Liang, Y. 2018. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems*, 31.

Lin, X.; Dai, Z.; Verma, A.; Ng, S.-K.; Jaillet, P.; and Low, B. K. H. 2024. Prompt Optimization with Human Feedback. *arXiv preprint arXiv:2405.17346*.

Liu, T.; Qin, Z.; Wu, J.; Shen, J.; Khalman, M.; Joshi, R.; Zhao, Y.; Saleh, M.; Baumgartner, S.; Liu, J.; et al. 2024. LiPO: Listwise Preference Optimization through Learning-to-Rank. *arXiv preprint arXiv:2402.01878*.

Morimura, T.; Sakamoto, M.; Jinnai, Y.; Abe, K.; and Air, K. 2024. Filtered Direct Preference Optimization. *arXiv preprint arXiv:2404.13846*.

Mukherjee, S.; Mitra, A.; Jawahar, G.; Agarwal, S.; Palangi, H.; and Awadallah, A. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

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

Peng, X. B.; Kumar, A.; Zhang, G.; and Levine, S. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*.

Perez, E.; Ringer, S.; Lukošiūtė, K.; Nguyen, K.; Chen, E.; Heiner, S.; Pettit, C.; Olsson, C.; Kundu, S.; Kadavath, S.; Jones, A.; Chen, A.; Mann, B.; Israel, B.; Seethor, B.; McKinnon, C.; Olah, C.; Yan, D.; Amodei, D.; Amodei, D.; Drain, D.; Li, D.; Tran-Johnson, E.; Khundadze, G.; Kernion, J.; Landis, J.; Kerr, J.; Mueller, J.; Hyun, J.; Landau, J.; Ndousse, K.; Goldberg, L.; Lovitt, L.; Lucas, M.; Sellitto, M.; Zhang, M.; Kingsland, N.; Elhage, N.; Joseph, N.; Mercado, N.; DasSarma, N.; Rausch, O.; Larson, R.; McCandlish, S.; Johnston, S.; Kravec, S.; El Showk, S.; Lanham, T.; Telleen-Lawton, T.; Brown, T.; Henighan, T.; Hume, T.; Bai, Y.; Hatfield-Dodds, Z.; Clark, J.; Bowman, S. R.; Askell, A.; Grosse, R.; Hernandez, D.; Ganguli, D.; Hubinger, E.; Schiefer, N.; and Kaplan, J. 2022. Discovering Language Model Behaviors with Model-Written Evaluations.

Peters, J.; and Schaal, S. 2007. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, 745–750.

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

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Shi, W.; Han, X.; Gonen, H.; Holtzman, A.; Tsvetkov, Y.; and Zettlemoyer, L. 2023. Toward Human Readable Prompt Tuning: Kubrick's The Shining is a good movie, and a good prompt too? In *Proc. EMNLP*, 10994–11005.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proc. EMNLP*, 4222–4235.

Sun, T.; He, Z.; Qian, H.; Huang, X.; and Qiu, X. 2022a. BBTv2: Pure Black-Box Optimization Can Be Comparable to Gradient Descent for Few-Shot Learning. In *Proc. EMNLP*, 3916–3930.

Sun, T.; Shao, Y.; Qian, H.; Huang, X.; and Qiu, X. 2022b. Black-box tuning for language-model-as-a-service. In *Proc. ICML*, 20841–20855.

Tang, Y.; Guo, Z. D.; Zheng, Z.; Calandriello, D.; Munos, R.; Rowland, M.; Richemond, P. H.; Valko, M.; Pires, B. Á.; and Piot, B. 2024. Generalized Preference Optimization: A Unified Approach to Offline Alignment. *arXiv preprint arXiv:2402.05749*.

Trivedi, P.; Chakraborty, S.; Reddy, A.; Aggarwal, V.; Bedi, A. S.; and Atia, G. K. 2025. Align-Pro: A Principled Approach to Prompt Optimization for LLM Alignment. *arXiv preprint arXiv:2501.03486*.

Wang, B.; Zheng, R.; Chen, L.; Liu, Y.; Dou, S.; Huang, C.; Shen, W.; Jin, S.; Zhou, E.; Shi, C.; et al. 2024. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Wang, C.; Jiang, Y.; Yang, C.; Liu, H.; and Chen, Y. 2023a. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. *arXiv preprint arXiv:2309.16240*.

Wang, X.; Li, C.; Wang, Z.; Bai, F.; Luo, H.; Zhang, J.; Jojic, N.; Xing, E. P.; and Hu, Z. 2023b. PromptAgent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*.

Wang, Z.; Dong, Y.; Zeng, J.; Adams, V.; Sreedhar, M. N.; Egert, D.; Delalleau, O.; Scowcroft, J. P.; Kant, N.; Swope, A.; et al. 2023c. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528*.

Wichers, N.; Denison, C.; and Beirami, A. 2024. Gradient-based language model red teaming. *arXiv preprint arXiv:2401.16656*.

Zhao, Y.; Joshi, R.; Liu, T.; Khalman, M.; Saleh, M.; and Liu, P. J. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Zhong, Z.; Friedman, D.; and Chen, D. 2021. Factual Probing Is [MASK]: Learning vs. Learning to Recall. In *Proc. NAACL*, 5017–5033.

Zhou, Y.; Muresanu, A. I.; Han, Z.; Paster, K.; Pitis, S.; Chan, H.; and Ba, J. 2023. Large Language Models Are Human-Level Prompt Engineers. In *Proc. ICLR*.

Zhu, B.; Jiao, J.; and Jordan, M. I. 2023. Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons. *arXiv preprint arXiv:2301.11270*.

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

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019b. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.