Optimizing Language Models with Fair and Stable Reward Composition in Reinforcement Learning

Jiahui Li[12]*, **Hanlin Zhang**[2]*, **Fengda Zhang**[1]*, **Tai-Wei Chang**[2]*
Kun Kuang[1]†, **Long Chen**[3], **Jun Zhou**[2]
[1]Zhejiang University, [2]Ant Group, [3]HKUST
jiahuil@zju.edu.cn, hanlinz144@gmail.com, fdzhang@zju.edu.cn,
taiwei.twc@antgroup.com, kunkuang@zju.edu.cn, longchen@ust.hk
jun.zhoujun@antgroup.com

Abstract

Reinforcement learning from human feedback (RLHF) and AI-generated feedback (RLAIF) have become prominent techniques that significantly enhance the functionality of pre-trained language models (LMs). These methods harness feedback, sourced either from humans or AI, as direct rewards or to shape reward models that steer LM optimization. Nonetheless, the effective integration of rewards from diverse sources presents a significant challenge due to their disparate characteristics. To address this, recent research has developed algorithms incorporating strategies such as weighting, ranking, and constraining to handle this complexity. Despite these innovations, a bias toward disproportionately high rewards can still skew the reinforcement learning process and negatively impact LM performance. This paper explores a methodology for reward composition that enables simultaneous improvements in LMs across multiple dimensions. Inspired by fairness theory, we introduce a training algorithm that aims to reduce *Disparity* and enhance *Stability* among various rewards. Our method treats the aggregate reward as a dynamic weighted sum of individual rewards, with alternating updates to the weights and model parameters. For efficient and straightforward implementation, we employ an estimation technique rooted in the mirror descent method for weight updates, eliminating the need for gradient computations. The empirical results under various types of rewards across a wide range of scenarios demonstrate the effectiveness of our method.

1 Introduction

In recent years, pretrained Language Models (LMs) have made significant strides in the field of natural language processing, leading to their widespread use in downstream applications such as conversational agents (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023; Xue et al., 2023), code generation (Ahmad et al., 2021; Wang et al., 2021; Roziere et al., 2023), and machine translation (Wang et al., 2023; Moslem et al., 2023). Reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Rafailov et al., 2023) and reinforcement learning from AI feedback (RLAIF) (Bai et al., 2022; Moskovitz et al., 2023; Lee et al., 2023; Havrilla et al., 2024) plays a critical role in this evolution, enhancing the models' ability to generate outputs that better align with human preferences and greatly increasing their versatility.

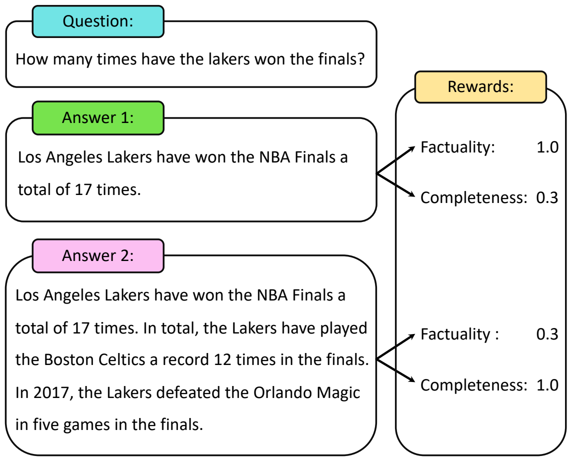


Figure 1: An example of question answering with two types of rewards optimizing in different directions.

The methods of RLHF and RLAIF typically incorporate three principal stages. Initially, there is supervised fine-tuning, which entails honing a foundational language model by utilizing a specialized dataset crafted for this purpose. Following this, the second stage is the development of reward functions, which are designed to serve as surrogate indicators of human or AI judgments and preferences. Subsequently, the language model serves as policy model and undergoes optimization via a reinforce-

*Equal contribution
†Corresponding Author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 10122–10140
November 12-16, 2024 ©2024 Association for Computational Linguistics

ment learning algorithm (Schulman et al., 2017). It is evident that the reward model is crucial in ensuring the language model's outputs continually improves and adapts to evaluative standards, which in turn, directly impacts the efficacy of the reinforcement learning phase. In some scenarios, there may be multiple reward functions (Ramamurthy et al., 2022; Glaese et al., 2022; Yuan et al., 2023; Bakker et al., 2022; Moskovitz et al., 2023), as people may wish to assess and enhance the LM from various perspectives. However, two factors make this challenging. Firstly, different reward functions evaluate text quality from various angles, but their insights are not entirely independent. Secondly, it is difficult for LMs to determine the specific optimization direction for each reward function since they only receive an aggregated reward score.

Considering the scenario depicted in Figure 1, where two reward functions are employed, let us examine the example question, "How many times have the Lakers won the finals?" The language model may generate various responses, which the two reward functions assess based on factuality and completeness, providing guidance for optimizing the language model. The first response is given a factuality score of 1.0 and a completeness score of 0.3, indicating that it is entirely factual but lacks some detail. The second response receives a factuality score of 0.3 and a completeness score of 1.0, signaling that it is complete but contains inaccuracies. It becomes challenging to evaluate the overall success of the responses by merely aggregating the individual rewards because they each excel in different aspects. Therefore, the development of an effective algorithm that can integrate various reward functions is of utmost significance for both research and practical applications. Recent studies have investigated multiple methods for composing rewards, such as ranking (Yuan et al., 2023), applying weightings (Wu et al., 2023), using welfare functions (Bakker et al., 2022), and practicing safe reinforcement learning (Moskovitz et al., 2023). While these approaches may involve complex designs and the fine-tuning of hyperparameters, there is a still high risk that the policy may overfocus on one reward function and neglect others, thereby negatively impacting the LM.

To tackle the aforementioned challenges, we propose a method named Fast RL (**Fa**ir and **St**able Reward **R**einforcement **L**earning), which is designed for simple but effective integration of diverse rewards. Inspired by fairness theory (Zhang et al., 2022; Ding et al., 2021), we have formulated a training objective that aims to minimize *Disparity* and maximize *Stability* across different reward functions simultaneously. Drawing on the principles of distributionally robust optimization (DRO) (Duchi and Namkoong, 2019; Wiesemann et al., 2014; Namkoong and Duchi, 2016; Zhang et al., 2022), we compute composite rewards as a weighted sum of individual rewards and transform the training objective into a max-min optimization problem. We iteratively optimize the language model and the weights, with the latter being updated via an estimation of the mirror descent method without the need for gradient computation. This strategy not only guides language models towards a more balanced, stable, and comprehensive improvement, but also offers simplicity in implementation.

Our contributions are summarized as follows:

- We present an algorithm that integrates various rewards during the reinforcement learning process, leading to a more comprehensive improvement of LMs.
- Our method is both simple and effective, allowing for easy adaptation to different types of reward functions or models without incurring significant computational overheads.
- We demonstrate the effectiveness of our approach through experimental results across various scenarios involving diverse rewards.

2 Preliminaries

2.1 Environments: Generation as MDP

Natural language generation can be conceptualized as a Markov Decision Process (MDP) (Puterman, 2014), represented as a tuple $\mathcal{M} \triangleq (\mathcal{S}, \mathcal{A}, \mathcal{R}, P, \gamma, T)$. At the start of each episode, a prompt input $x = (x_0, x_1, ..., x_m)$ is sampled from the data buffer, serving as the initial state s_0, with $s_0 \in \mathcal{S}$, $x_m \in \mathcal{V}$, where $\mathcal{S}$ denotes the state space and $\mathcal{V}$ denotes a finite vocabulary. At every timestep t, the language model functions as a policy $\pi(a_t|s_t)$, generating a token that signifies choosing an action $a_t \in \mathcal{A}$ based on its current state s_t. A new state is subsequently reached via the transition function $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. An episode concludes when the timestep exceeds the maximum horizon length T or when an end-of-text (EOT) token is produced. The generated response is denoted by $y =$

$(a_0, a_1, ..., a_T)$. Summatively, an episode is captured as a trajectory $\tau = (s_i, a_0, ..., s_T, a_T)$, with the policy model's objective being to maximize the expected return $R(\tau) = \sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t)$, where $\mathcal{R} \in \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ represents the reward function and $\gamma \in [0, 1)$ symbolizes the discount factor.

2.2 Reward Functions for Optimizing the Language Models

Reward functions can be broadly divided into two main categories. The first category (Bakker et al., 2022; Yuan et al., 2023; Wu et al., 2023; Rafailov et al., 2023) consists of trained models that act as proxies for human preferences within specific contexts, typically using Bradley-Terry models (Bradley and Terry, 1952). The second category (Ramamurthy et al., 2022; Moskovitz et al., 2023) includes commonly used metrics in NLP, such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). These metrics do not require the reward model training procedure, thus allowing for automatic measurement and quick implementation. In addition to these categories, rewards can be classified as either coarse-grained or fine-grained. Coarse-grained rewards provide a single, sparse reward at the end of each episode, reflecting the overall success of the generation. Fine-grained rewards, in contrast, can be assigned for each token or subsentence, reflecting success across a range of timesteps. In situations where different types of rewards coexist, we define the composite reward function as follows:

$$r_{com} = f(r_1, ..., r_n), \tag{1}$$

where r_i represents the output reward from various reward functions, n is the number of reward functions involved, and $f(\cdot)$ is any composite function. For simplicity, the timestep subscript t is omitted here and will continue to be excluded in the remainder of the text.

3 Method

In this section, we propose a method and its implementation that are designed to simultaneously improve the performance of LMs across various reward functions. The entire training framework is depicted in Figure 2. Initially, the input state is fed into the LM, and each reward function provides a score for evaluating the model outputs. These scores are then integrated using a weighted sum to obtain a composite reward, which is subsequently used to optimize the LM. Concurrently, the weights are updated through an estimation based on mirror descent.

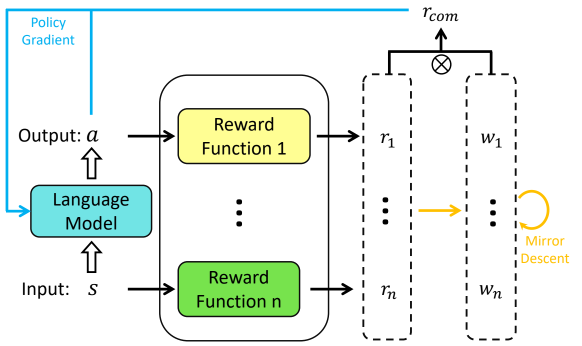


Figure 2: Training framework of Fast RL. The parameters of LM are updated using policy gradient, while the weights of different rewards are adjusted through mirror descent.

3.1 Optimization Objective

Drawing inspiration from fairness theory, our goal is to train a LM that achieves minimal *Disparity* and maximal *Stability* across various reward functions simultaneously. We define this objective as follows:

$$\begin{aligned} \text{Obj}_{\pi_\theta} &:= \inf_{r_{com}} \mathbb{E}_{(s,a)\sim D_b}[r_{com}(\pi_\theta, (s, a))]. \\ r_{com} &:= \{\sum_{i=1}^{n} w_i r_i | \sum_{i=1}^{n} w_i = 1, w_i \geq 0\}. \end{aligned} \tag{2}$$

where n denotes the number of the reward functions, D_b denotes the samples from the replay buffer, $r_i = \mathcal{R}(s, a)$ denotes the rewards output by i-th reward function or model $\mathcal{R}$, w_i denotes the weights of r_i, r_{com} denotes the composite reward, and π_θ denotes the language model with parameter θ.

3.2 Simple yet Effective Implementation

Borrowing the idea from distributionally robust optimization (DRO) (Duchi and Namkoong, 2019; Wiesemann et al., 2014; Namkoong and Duchi, 2016; Zhang et al., 2022), the problem of maximizing the objective in Equation (2) can be rewritten as:

$$\max_{\theta} \min_{\sum_{i=1}^{n} w_i = 1, w_i \geq 0} \sum_{i=0}^{n} w_i r_i(\pi_\theta, (s, a)). \tag{3}$$

To address this optimization problem, we can alternatively optimize the policy parameters π and the weights w_i. Rather than employing gradient descent, we utilize an estimation technique inherent to the mirror descent method for updating w_i. Each w_i is updated via:

$$w_i^{\text{cur}} = \frac{w_i^{\text{pre}} \exp(-\lambda r_i)}{\sum_{j=1}^{n} w_j^{\text{pre}} \exp(-\lambda r_j)}, \tag{4}$$

where pre denotes the previous update step, cur denotes the current update step, and λ is a hyper-parameter. Notably, we initial $w_i^{\text{pre}} = \frac{1}{n}$ in the first iteration of our experiment.

In some scenarios, rewards may conflict. We address this by smoothing rewards and incorporating a bias factor, as shown in Equation (5):

$$w_i^{\text{cur}} = \frac{w_i^{\text{pre}} \exp(-\lambda \text{smooth}(r_i) + b_i)}{\sum_{j=1}^{n} w_j^{\text{pre}} \exp(-\lambda \text{smooth}(r_j) + b_j)}, \tag{5}$$

where b_i is a bias factor from prior knowledge, and smooth is a smoothing function. This function reduces the influence of extreme rewards, preventing overreaction to single signals and balancing multiple objectives. Meanwhile, the bias factor allows prioritization of certain objectives, guided by prior knowledge, to resolve conflicts and maintain focus on critical goals.

The composite reward is computed as.

$$r_{com} = \sum_{i=1}^{n} w_i^{\text{cur}} r_i. \tag{6}$$

However, there is a risk of over-optimization, a phenomenon where maximizing returns on the reward function beyond a certain threshold could actually reduce the performance of the policy model. In line with recent studies, we incorporate a composite reward with a KL penalty to moderate the policy model's propensity for over-optimization (Ramamurthy et al., 2022; Moskovitz et al., 2023; Wu et al., 2023):

$$r_{exp} = r_{com} - \beta \cdot \text{KL}(\pi_\theta(a|s) \parallel \pi_{ref}(a|s)), \tag{7}$$

where π_θ represents the policy model, π_{ref} indicates the reference model, and β is the coefficient that controls the strength of the KL penalty. This adjusted reward r_{exp} can be used to fine-tune the language model using any reinforcement learning algorithm, and in this paper, we select Proximal Policy Optimization (PPO) (Schulman et al., 2017). The details of our implementation are provided in Section4.

3.3 Analysis

In this subsection, we offer a theoretical analysis of our proposed method.

Theorem 1. *Let $r_i := \mathbb{E}_{(s,a)\sim D_b}[r_i(\pi_\theta, (s,a))]$ be an expectation of reward in dataset D_b, $\mathbf{w} \in \Delta^{n-1}$ be the group weights, n be the total number of the reward functions, $\bar{r}^u$ be the average of the rewards, $d_i := (r_i - \bar{r})^2$ and $Var(r_i) := \frac{1}{n}\sum_{i=1}^{n} d_i$ be the variance of rewards. If $\min_i\{\frac{d_i}{\sum_{i=1}^{n} d_i}\} \geq \frac{1}{\|n\mathbf{w}-\mathbf{1}\|_2^2}$, then there exists a constant $C > 0$ such that*

$$Obj_{\pi_\theta} = \bar{r} - C\sqrt{Var_{i\in[n]} r_i}. \tag{8}$$

The theorem demonstrates that our objective, Obj_{π_θ}, can be interpreted as a combination of the average score across the reward functions and a regularization term that mitigates over-focus on any single reward function, particularly when the reward distribution is imbalanced. This design ensures that the language model maintains balanced performance across all reward functions, preventing excessive reliance on any one of them. For poof and more details, please refer to Appendix.

3.4 Training Algorithm

The comprehensive training protocol we adopted is encapsulated in Algorithm 1. This framework adheres to the standard Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017), augmented with additional steps dedicated to the calculation of the composite reward and the update of the weights.

4 Experiment

We evaluate the effectiveness of our method across various scenarios using different language models. Our experiments encompass dialogue generation, question answering, and tasks aimed at mitigating harmfulness and enhancing helpfulness. Our proposed method consistently outperforms the baselines across all experimental scenarios. Details regarding hyperparameters are provided in the Appendix.

4.1 Dialogue Generation

4.1.1 Experimental Settings

Dataset. We conducted an experiment utilizing the widely recognized DailyDialog dataset (Li et al., 2017), consisting of transcripts from human conversations.

Algorithm 1 Optimizing a Language Model with Multiple Reward Models

Initialize: reference language model π_{ref}; initial value model V_φ; n reward models $\mathcal{R}_1, ..., \mathcal{R}_n$; initial weights $w_1^{\text{pre}}, ..., w_n^{\text{pre}}$;task dataset D; hyperparameters

Finetune the reference language model on dataset D and get the initial policy model π_θ
Training the reward models $\mathcal{R}_1, ..., \mathcal{R}_n$ on dataset D
Training the composition reward model f on dataset D
for epoch $ep = 1, ..., k$ **do**
    Sample a batch D_b from D
    Sample output sequence $y^i \sim \pi_\theta(\cdot|x^i)$ for each $x^i \in D_b$
    Compute rewards $r_1, ..., r_n$ via $\mathcal{R}_1, ..., \mathcal{R}_n$
    Compute composite rewards r_{com} via Equation (4) and Equation (6)
    Compute penalized rewards r_{exp} via Equation (7)
    Set $w_i^{\text{pre}} = w_i^{\text{cur}}$ for each i
    Compute advantages $\{A\}_{t=1}^{|y^i|}$ and target values $\{V'\}_{t=1}^{|y^i|}$ for each y^i with V_φ
    Update the policy model by:
    $\theta \leftarrow \arg\max_\theta \frac{1}{|D_b|} \sum_{i=1}^{D_b} \frac{1}{|y^i|} \sum_{t=1}^{y_i} \min(\frac{\pi_\theta(a_t|s_t)}{\pi_{ref}(a_t|s_t)} A_t, \text{clip}(\frac{\pi_\theta(a_t|s_t)}{\pi_{ref}(a_t|s_t)}, 1-\epsilon, 1+\epsilon) A_t)$
    Update the value model by:
    $\varphi \leftarrow \arg\min_\varphi \frac{1}{|D_b|} \sum_{i=1}^{D_b} \frac{1}{|y^i|} \sum_{t=1}^{y_i} (V_\varphi(a_t|s_t) - V'(a_t|s_t))^2$
end for

Output: π_θ

Reward Functions. Following Moskovitz et al. (2023), we selected METEOR (Banerjee and Lavie, 2005), Intent Score (Ramamurthy et al., 2022), BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2019) as reward functions. These models capture the desired behavior of text from different perspectives and can be implemented quickly. Among these, Intent Score and BERTScore are estimated using a pretrained human preference model, RoBERTa (Liu et al., 2019), and BERT (Devlin et al., 2018), respectively, while the other two utilize n-gram metrics. The reward scores are provided at the end of each response to reflect its overall quality.

Baselines. In our study, we utilized GPT-2 (Radford et al., 2019) as the starting point for our policy model. For our baseline algorithm, we chose Proximal Policy Optimization (PPO) (Schulman et al., 2017), within which rewards are calculated through a linear combination of individual metrics, each metric assigned a fixed weight that is predetermined. Furthermore, we incorporated Constrained Reinforcement Learning (Constrained RL) (Moskovitz et al., 2023) to serve as an additional baseline for comparison. Constrained RL uses the Nelder-Mead method (Nelder and Mead, 1965) to iteratively update a simplex of thresholds based on reward evaluation performance, with parameters $\alpha = 1$ $\gamma = 1$, $\rho = 0.5$, $\sigma = 1$.

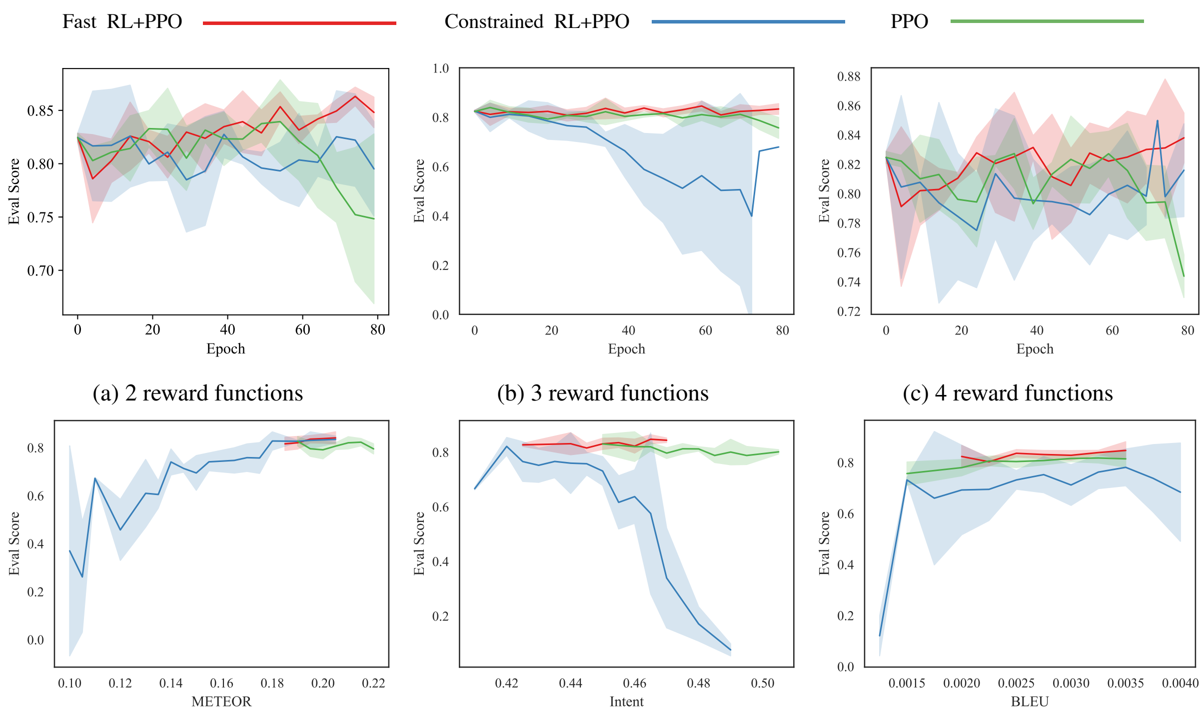
Evaluation Metrics. We assessed various methods based on two criteria. (1) In an approach similar to that of Moskovitz et al. (2023), we computed an **evaluation score using six distinct metrics**. These metrics, as identified by Moskovitz et al. (2023), operate independently of the reward functions. Specifically, we select SacreBLEU (m_a) (Post, 2018), ROUGE-2 (m_b) (Lin, 2004; Ganesan, 2018), and ROUGE-L (m_c) as metrics related to lexicon, and Conditional Entropy-3 (m_u), vocab-size-3-nopunct (m_v), and mean-prediction-length-nopunct (m_w) as metrics related to diversity. We normalized the score of each metric to fall within a range of 0 to 1, using the minimum and maximum values observed in Constrained RL experiments across three distinct reward function settings. The evaluation score (m_{eval}) is subsequently computed as outlined in Equation (9)

$$m_{eval} = \frac{m_a + m_b + m_c + m_u + m_v + m_w}{6}. \tag{9}$$

(2) We adopt **GPT-4** (Achiam et al., 2023) **as an proxy for human judgment** to further assess the methods.

4.1.2 Experimental Results

Stability Across Varying Numbers of Reward. We conducted experiments utilizing configurations with 2, 3, and 4 reward functions. Figure 3(a-c)



(a) 2 reward functions (b) 3 reward functions (c) 4 reward functions

(d) 3 reward functions (e) 3 reward functions (f) 3 reward functions

Figure 3: The evaluation score of different methods across three scenarios with varying number of rewards.

illustrates the improvement in model performance over the training epochs, where the results represent the mean of three random seeds, and the shaded area indicates the standard deviation. In the initial epochs of training, all methods remained stable when only two reward functions, METEOR and Intent Score, were used. However, the incorporation of a third reward function, BLEU, resulted in a significant deterioration in the performance of Constrained RL, making the training unstable. In contrast, both the baseline PPO and our method demonstrated stability. Upon introducing another reward function, BERTScore, only our method maintained stability. Moreover, our method outperformed the baselines in evaluation score across all scenarios, thereby demonstrating the effectiveness of the composite reward. The strong baseline, Constrained RL, delivered unsatisfactory performance, except in the two-reward configuration. This can be primarily attributed to its explicit requirement for rewards from each aspect to surpass certain thresholds, without considering potential conflicts among them. Therefore, the type and number of reward functions employed significantly influences performance, highlighting the importance of carefully selecting and harmonizing reward functions to achieve balanced and optimal training outcomes.

Overoptimization and Reward Conflict Phenomena. Nonetheless, two phenomena require attention. Firstly, the performance of the language model tends to decline after approximately 75 epochs, which may be due to the fact that KL regularization, despite mitigating optimization, cannot completely eliminate it. Consequently, there is a tendency for the policy to overfit on the reward functions. Secondly, the peak performance obtained with three reward functions is lower than that achieved with two, a result that may stem from the potential conflict between differing objectives, impairing further improvement.

Improving the LM Comprehensively. Figure 3(d-f) illustrates the variation in evaluation scores across different metrics in the 3-reward-model setting. It is evident that our method exhibits a much tighter distribution of each reward score and experiences less fluctuation over the course of training, ultimately achieving the highest evaluation score. This further demonstrates both the effectiveness and the stability of our method.

GPT4 Evaluation. To objectively validate the efficacy of different methods, we conducted an evaluation using GPT-4 (OpenAI, 2023) as a proxy for human judgment. We randomly sampled 50 dialogue contexts from the dataset, along with their gener-

Table 1: GPT-4 evaluation results on DailyDialog.

Method	Selection Rate
PPO	10%
ConstrainedRL+PPO	22%
Fast RL+PPO(Ours)	**66%**
No preference	2%

ated responses, for this evaluation. The task for GPT-4 was to select the most appropriate response given the context. Moreover, we allowed GPT-4 the option to choose "no preference" in cases where it encountered difficulty in discerning a clear favorite, or if none of the responses seemed fitting. As illustrated in Table 1, our method achieves the highest selection rate, confirming that it significantly outperforms competing approaches in terms of performance. We have provided the GPT-4 prompts and examples of the showcases in the Appendix.

4.2 Question Answering

4.2.1 Experimental Settings

Dataset. We conduct experiment on QAFeedback dataset provided by Wu et al. (2023), consisting of 3,853 training examples, 500 development examples, and 948 test examples.

Reward Models. In this scenario, three reward models are trained, each focusing on a specific category: relevance, correctness, and completeness. Notably, only the completeness reward model is the Bradley-Terry (Bradley and Terry, 1952) model. These models evaluate the response and assign reward scores to each sub-sentence, ensuring a thorough assessment across the crucial aspects of the text.

Baseline. Following Wu et al. (2023), we selected T5-large (Raffel et al., 2020) as the base model and fine-tuned it with 1,000 training examples to develop the SFT model. We consider this model as the baseline and also use it as the initial policy model for RL. We also compared our method to Fine-grained RL (F.G. RL) (Wu et al., 2023), which combines different rewards using fixed expert-defined weights. Following the original paper, we set the weights for relevance, factuality, and completeness rewards at 0.3, 0.5, and 0.3, respectively.

4.2.2 Experimental Results

Reward Model Evaluation. We employ the trained reward models to evaluate the responses generated by various methods. The results for the test dataset are presented in Table 2, where R_1, R_2, and R_3 denote the relevance reward, factuality reward, and completeness reward, respectively. Compared to baseline methods, our approach achieves the maximum reward in nearly all aspects, with the exception of factuality. This discrepancy can be attributed to the inherent conflicts among these reward models, making simultaneous optimization challenging (please refer to the Appendix for more details).

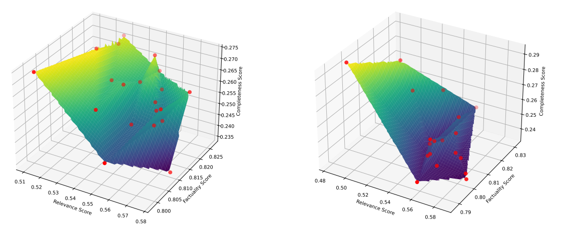
Table 2: Results on QAFeedback test set.

Method	Rouge	R_1	R_2	R_3
SFT	49.16	0.469	0.793	0.225
F.G. RL	50.16	**0.518**	**0.823**	0.226
Fast RL	**50.28**	**0.518**	0.822	**0.243**

GPT-4 Evaluation. Similar to previous works (Rafailov et al., 2023; Dai et al., 2023), we randomly selected 50 test examples and asked GPT-4 to comprehensively evaluate the quality of the response, considering all three aspects simultaneously. We present the win rate in comparison to the SFT model, with the results shown in Table 3. Our method improved the win rate by approximately 8% and significantly reduced the lose rate, from 22% to 8%.

Table 3: GPT-4 Evaluation on QAFeedback test set.

vs. SFT	Win	Tie	Lose
F.G RL	22%	56%	22%
Fast RL	**30%**	62%	**8%**



(a) F.G. RL (b) Fast RL

Figure 4: Correlations among different reward models in QAFeedback.

Visualizing the Rewards. To more effectively analyze the correlations among the various rewards in the question answering task, we plotted the reward values at each timestep and fitted a polynomial surface to the data, as shown in Figure 4. It is evident that the reward for relevance conflicts with the other two types of rewards, complicating the optimization of the policy. Our proposed method exhibits a more compact and concentrated distribution compared to the baseline, demonstrating that we focus on different rewards simultaneously, which leads to a more stable and comprehensive improvement of the LM.

4.3 Harmfulness Mitigation&Helpfulness Enhancement

4.3.1 Experimental Settings

Dataset. We conduct experiment on Alpaca (Taori et al., 2023) and SafeRLHF (Dai et al., 2023) datasets. The former is used to supervised fine-tuning the language model while the latter is used to train the reward model and perform reinforcement learning.

Reward Models. Following Dai et al. (2023), we train two Bradley-Terry models (Bradley and Terry, 1952) to predict the rewards and costs of a generated sentence.

Baseline. We select the LLaMA-7B (Touvron et al., 2023) model as the base model. Meanwhile, we adopt reward shaping(R.S) (Ng et al., 1999) as the baseline. For the reward shaping approach, the composite reward, excluding the KL penalty, is calculated as $r_{com} = \frac{1}{2} \times (\mathcal{R}_\phi(x, y) + \alpha \times \mathcal{C}_\varphi(x, y))$, where $\mathcal{R}$ denotes the reward model, $\mathcal{C}$ denotes the cost model, and α is the scaling factor which is set to -1 in our experiments.

Evaluation Metrics. Our experimental evaluation is conducted using two distinct methods: (1) Reward Evaluation. This involves two sub-criteria: (a) We compare the average reward and cost scores within the test set. (b) We assess the win rate for helpfulness (measured by a higher reward score compared to the SFT model) and the rate of safe responses (costs being lower than 0) to gauge the practical utility and safety of the responses. (2) GPT-4 Evaluation (Achiam et al., 2023). We assess the win rates of various methods against the SFT model by employing GPT-4 as a stand-in for human evaluators.

4.3.2 Experimental Results

Reward Model Evaluation. The results are presented in Table 4. When compared to the R.S with fixed weights, Fast RL achieves higher rewards and incurs lower costs, which highlight the efficacy of our method.

Table 4: Reward evaluation of the SafeRLHF test set.

Metric	Reward	Cost	Helpful Win Rate	Safe Rate
R.S	1.818	0.916	68.75%	44.26%
Fast RL	**1.906**	**0.894**	**71.27%**	44.89%

Table 5: GPT-4 evaluation of the SafeRLHF test set.

vs. SFT	Win	Tie	Lose
R.S	17.0%	63.5%	19.5%
Fast RL	**20.5%**	61.0%	18.5%

GPT-4 Evaluation. We prompt GPT-4 with assessing the harmlessness, helpfulness, and level of detail in the generated responses, with an emphasis on harmlessness as the top priority. The comparative win rates against the SFT model are presented in Table 5. The baseline method, R.S, achieves a lower win rate when compared to the SFT model. This can be attributed to the fact that the fixed weights in R.S cause it to excessively concentrate on maximizing rewards while disregarding the costs, which can result in more harmful responses from the language model. In contrast, our method considers both rewards and costs simultaneously, leading to responses that are not only better but also safer.

5 Related Work

Reinforcement Learning for Optimizing the Language Model. RLHF (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Rafailov et al., 2023) has emerged as a crucial methodology for fine-tuning language models to better reflect human intentions, as documented in various studies. Its efficacy is demonstrated in downstream tasks such as summarization (Stiennon et al., 2020), story-telling (Ziegler et al., 2019), following instructions, and reducing harm (Bai et al., 2022; Lu et al., 2022; Ganguli et al.). However, RLHF involves gathering pairwise human-labeled data and an additional training procedure for the reward model, which can be resource-intensive. To optimize LMs in a faster and more lightweight manner, recent studies have shifted toward applying Reward Learning from AI Feedback (RLAIF) (Bai et al., 2022; Moskovitz et al., 2023; Lee et al., 2023; Havrilla et al., 2024). This approach leverages AI-generated feedback or provides direct reward signals, thus bypassing the need for extensive human-labeled datasets. Furthermore, research by Li et al. (2023) has found that reward-model-based approaches continue to hold their benefits, particularly when dealing with samples that are not well represented within the initial training preferences. These insights underscore the sustained importance of reinforcement learning in the enhancement of language models.

Integrating Diverse Rewards. To enhance the language model's alignment with diverse preferences, various forms of feedback are typically utilized to reflect the policy's behavior across multiple dimensions (Bakker et al., 2022; Glaese et al., 2022; Yuan et al., 2023; Wu et al., 2023; Moskovitz et al., 2023). Integrating disparate rewards, however, presents a significant challenge, as the policy may struggle to discern the intentions behind the rewards' design, receiving feedback in the form of a single scalar value. Traditional studies (Wu et al., 2023; Ramamurthy et al., 2022) have attempted to address this by aggregating the different rewards and assigning predefined weights based on prior knowledge. In contrast, a separate line of research (Yuan et al., 2023; Glaese et al., 2022) recommends policy optimization through the ranking of multiple sampled responses. More specifically, Yuan et al. (2023) developed a ranking loss that increases the likelihood of selecting higher-quality responses, and Glaese et al. (2022) introduced a reranking score to act as the overall reward, rewarding the higher-quality responses among a set of samples. Further, Bakker et al. (2022) suggested a welfare function that measures and orders consensus statements by their desirability to combined reward models. Moreover, Moskovitz et al. (2023) implemented constrained reinforcement learning to prevent the agent from over-optimizing individual reward models beyond certain thresholds. Despite these innovative approaches, the risk remains that policy models may give undue emphasis to certain individual rewards. Therefore, we are exploring a method that leverages fairness theory to yield an anticipated reward that holistically enhances the language model. This approach aims to balance the multiple objectives and reflect a fair distribution of attention across the varying rewards, ensuring a more equitable and effective improvement of the language model.

6 Conclusion

In this study, we focus on the scenarios that involve complex, multi-faceted reward models for optimizing LMs. Given the diverse perspectives from which various reward models assess text, our aim is to develop a method that can appropriately compose different rewards, so as to ensure that LMs do not excessively prioritize one perspective over others. Leveraging fairness theory, we propose a method wherein the training objective is to reduce disparity and increase robustness among rewards. Drawing on the principles of DRO, we calculate composite rewards as a weighted sum of individual rewards, and transforms the training objective into a max-min optimization problem. The updating mechanism for the weights assigned to different rewards utilizes an estimation approach based on the mirror descent method, which is not only straightforward but also highly effective, simplifying the implementation process. The empirical results across various scenarios demonstrate the efficacy of our approach.

Limitations and Future Work. While our study yields promising results, it is not without its limitations. Firstly, the absence of human evaluations is noteworthy; we have relied on GPT-4 as a stand-in, but this may not reflect human judgment with complete accuracy. Furthermore, the potential for conflict and inaccuracy arises from using outputs of various reward models, as our current approach does not have a mechanism to distinguish between the efficacy of these models. Instead, we calculate a composite reward in an effort to concurrently boost performance across all reward signals, which unfortunately may lead to less-than-ideal outcomes. In our future research, we plan to enhance our methodology by developing and implementing theoretical frameworks designed to detect and eliminate superfluous rewards.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62441605, 62376243), National Key Research and Development Program of China (2022YFC3340900), the Starry Night Science Fund at Shanghai Institute for Advanced Study (Zhejiang University), the Fundamental Research Funds for the Central Universities (226-2024-00170), and Ant Group Postdoctoral Programme. Long Chen is supported by HKUST Special Support for Young Faculty (F0927) and HKUST Sports Science and Technology Research Grant (SSTRG24EG04). All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. Unified pre-training for program understanding and generation. *arXiv preprint arXiv:2103.06333*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. 2022. Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, 35:38176–38189.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe rlhf: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34:6478–6490.

John Duchi and Hongseok Namkoong. 2019. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(68):1–55.

Kavita Ganesan. 2018. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*.

D Ganguli, A Askell, N Schiefer, T Liao, K Lukošiūtė, A Chen, et al. The capacity for moral self-correction in large language models. arxiv 2023. *arXiv preprint arXiv:2302.07459*.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2023. Remax: A simple, effective, and efficient method for aligning large language models. *arXiv preprint arXiv:2310.10505*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609.

Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D Dragan, and Stephen McAleer. 2023. Confronting reward model overoptimization with constrained rlhf. *arXiv preprint arXiv:2310.04373*.

Yasmin Moslem, Rejwanul Haque, and Andy Way. 2023. Adaptive machine translation with large language models. *arXiv preprint arXiv:2301.13294*.

Hongseok Namkoong and John C Duchi. 2016. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29.

John A Nelder and Roger Mead. 1965. A simplex method for function minimization. *The computer journal*, 7(4):308–313.

Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287.

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. *arXiv preprint arXiv:2304.02210*.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8696–8708.

Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. 2014. Distributionally robust convex optimization. *Operations research*, 62(6):1358–1376.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*.

Siqiao Xue, Caigao Jiang, Wenhui Shi, Fangyin Cheng, Keting Chen, Hongjun Yang, Zhiping Zhang, Jianshan He, Hongyang Zhang, Ganglin Wei, Wang Zhao, Fan Zhou, Danrui Qi, Hong Yi, Shaodong Liu, and Faqiang Chen. 2023. Db-gpt: Empowering database interactions with private large language models. *arXiv preprint arXiv:2312.17449*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Fengda Zhang, Kun Kuang, Yuxuan Liu, Long Chen, Jiaxun Lu, Fei Wu, Chao Wu, Jun Xiao, et al. 2022. Towards multi-level fairness and robustness on federated learning. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Experimental Details

A.1 Experimental Settings

Dialogue Generation. We adopted a similar experimental setup to that described by Ramamurthy et al. (2022); Moskovitz et al. (2023) for our dialogue generation, utilizing a context window spanning five utterances. Following Moskovitz et al. (2023), inputs to the model were presented as concatenated segments of human dialogue, with speaker transitions denoted by a distinct end-of-utterance (<EOU>) token. Additionally, the intent classification reward mechanism was established based on a fine-tuned RoBERTa (Liu et al., 2019) model. This system assigned a score of 1 when the model's inferred intent for a generated utterance matched that of the corresponding reference or ground-truth utterance, otherwise attributing a score of 0. Consistent with (Moskovitz et al., 2023), we adopted the GPT-2 (Radford et al., 2019) architecture for both the policy and value models. We selected four distinct rewards for our experiments, the specifics of which are detailed in Table 6.

Table 6: Chosen rewards in dialogue generation task.

Setting	Chosen Reward Function
2 rewards	METEOR; INTENT
3 rewards	METEOR; INTENT; BLEU
4 rewards	METEOR; INTENT; BLEU; BERT

Question Answering. In our question-answering scenario, diverging from our previous task, we opted for T5-large (Raffel et al., 2020) as the policy model and T5-base as the value model. We adopted the same reward models as those detailed by Wu et al. (2023), focusing on factuality, coherence, and completeness, with only the completeness reward model being a Bradley-Terry (Bradley and Terry, 1952) model. For more details, readers are directed to the original publication by Wu et al. (2023).

Harmfulness Mitigation&Helpfulness Enhancement. In the scenario of harm mitigation and helpfulness enhancement, we have adopted LLaMA-7B (Touvron et al., 2023) as both the policy model and the reward model. Following Dai et al. (2023), we utilize the Alpaca dataset for SFT and employ the SafeRLHF dataset for training both the reward and cost models.

Hyperparameters. We implement our algorithm in different benchmarks separately[1][2][3]. For transparency and reproducibility, we have detailed all the hyperparameters associated with fine-tuning the policy in Table 7,Table 8. and Table 9.

Table 7: RL Hyperparameters for DailyDialog and QAFeedback.

Settings	DailyDialog	QAFeedback
Total epochs	80	10
Batch size	64	12
Learning rate	1e-6	1e-5
Clip ratio ϵ	0.2	0.2
Rollouts top-k	20	20
Temperature	0.7	0.7
Discount factor γ	0.99	0.99
GAE λ	0.95	0.95
KL coefficient β	0.2	0.3
Policy model	GPT2	T5-large
Value model	GPT2	T5-base

Table 8: SFT and RM hyperparameters for harmfulness mitigation&helpfulness enhancement task.

Settings	SFT	RM
Dataset	Alpaca	SafeRLHF
Total epochs	3	2
Batch size per GPU	4	16
Learning rate	2e-5	2e-5
Lr warm up ratio	0.03	0.03
Lr scheduler type	Cosine	Cosine
Max length	512	512
Gradient acc steps	8	1
Weight decay	0.0	0.1
Bf16	TRUE	TRUE
Tf32	TRUE	TRUE

Pseudo Code. The proposed method is straightforward to implement and can be seamlessly integrated into any mainstream framework. The pseudo code is presented below.

Computational resources. Our experiments of dialogue generation and question answering were conducted on a single NVIDIA A100 GPU. For the dialogue generation task, the optimization of the language model typically required between 8 and 10 hours. For the question answering task, the optimization of the language model required between 25 and 30 hours. The Harmfulness Mitigation&Helpfulness experiment were conducted

[1] https://github.com/tedmoskovitz/ConstrainedRL4LMs
[2] https://github.com/allenai/FineGrainedRLHF
[3] https://github.com/PKU-Alignment/safe-rlhf

```
# init
num_rewards = n
self.w_list = [1/n for i in range(num_rewards)]
self.lambda_value = lambda_value

# mirror decent
temperature = 1.0
frac_base = torch.tensor(0.0)
composed_reward = torch.tensor(0.0)
rewards_list = get_reward_from_env()
for i in range(num_rewards):
    self.w_list[i] *= \
        torch.exp(self.lambda_value * rewards_list[i] / temperature)
    frac_base += self.w_list[i]
for i in range(num_rewards):
    self.w_list[i] = self.w_list[i]/frac_base
    composed_reward += self.w_list[i] * rewards_list[i]

# RL using composed_reward
......
```

Table 9: RL hyperparameters for SafeRLHF dataset.

Settings	SafeRLHF
Total epochs	3
Batch size per GPU	16
Num return sequences	1
Actor learning rate	1e-5
Actor Weight decay	0.01
Actor lr warm up ratio	0.03
Actor lr scheduler type	Cosine
Critic Learning rate	5e-6
Critic Weight decay	0.0
Critic lr warm up ratio	0.03
Critic lr scheduler type	Cosine
Clip ratio ϵ	0.2
Rollouts top-k	1
Temperature	1.0
Ptx coeff	16
GAE γ	1
GAE λ	0.95
Rf16	TRUE
Tf32	TRUE

on 8 NVIDIA A100 GPUs. he SFT procedure necessitates about 3 hours. Training both the reward and the cost model each requires about 14 hours, and the reinforcement learning phase takes approximately 10 hours.

B Theorems

B.1 Proof of Theorem 1

proof. Recall that our objective in a group of reward function is defined as:

$$\text{Obj}_{\pi_\theta} := \inf_{r_{com}} \mathbb{E}_{(s,a)\sim D_b}[r_{com}(\pi_\theta, (s,a))].$$
$$r_{com} := \{\sum_{i=1}^{n} w_i r_i | \sum_{i=1}^{n} w_i = 1, w_i \geq 0\}. \quad (10)$$

Borrowing techniques from distributional robustness optimization (Duchi and Namkoong, 2019; Wiesemann et al., 2014; Namkoong and Duchi, 2016; Zhang et al., 2022), the problem of maximizing the risk in Equation (10) can be rewritten as:

$$\max_{\theta} \min_{w_i} \sum_{i=0}^{n} w_i r_i(\pi_\theta, (s,a)),$$
$$s.t. \sum_{i=1}^{n} w_i = 1, w_i \geq 0. \quad (11)$$

Inspired by Duchi and Namkoong (2019), we introduce an instrumental variable $\mathbf{u}$ defined as:

$$\mathbf{u} := \mathbf{w} - \frac{1}{n}\mathbf{1}, \quad (12)$$

where $\mathbf{w} = (w_1, ..., w_n)$ and $\mathbf{u} = (u_1, ..., u_n)$. Then the objective function of Equation (11) can be rewritten as:

$$\begin{aligned}&\sum_{i=1}^{n} w_i r_i \\ =&\sum_{i=1}^{n} u_i r_i + \frac{1}{n}\sum_{i=1}^{n} r_i \\ =&\sum_{i=1}^{n} u_i r_i + \bar{r} \\ =&\sum_{i=1}^{n} u_i (r_i - \bar{r}) + \bar{r} \qquad (13) \\ =&-\sum_{i=1}^{n} u_i (\bar{r} - r_i) + \bar{r}. \qquad (14)\end{aligned}$$

Using Cauchy–Schwarz inequality, we have:

$$\begin{aligned}&-\sum_{i=1}^{n} u_i(\bar{r} - r_i) + \bar{r} \\ \geq&-\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} (r_i - \bar{r})^2} + \bar{r} \\ =&\bar{r} - \sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\mathrm{Var}_{i\in[n]} r_i}. \qquad (15)\end{aligned}$$

The equality can be obtained if and only if:

$$u_i = \sqrt{\frac{||\mathbf{u}||_2^2}{\sum_{i=1}^{n} d_i}} \cdot (r_i - \bar{r}). \qquad (16)$$

Recall that $\mathbf{u} := \mathbf{w} - \frac{1}{n}\mathbf{1}$, which requires that $\forall i$,

$$u_i = \sqrt{\frac{||\mathbf{u}||_2^2}{\sum_{i=1}^{n} d_i}} \cdot (r_i - \bar{r}) \geq -\frac{1}{n}, \qquad (17)$$

If $\min_i\{\frac{d_i}{\sum_{i=1}^{n} d_i}\} \geq \frac{1}{||n\mathbf{w}-\mathbf{1}||_2^2}$, then $\forall i$, we have equation (17) holds. This completes our proof.

B.2 Theorem 2 and Proof

Theorem 2. *Let $r_i := \mathbb{E}_{(s,a)\sim D_b}[r_i(\pi_\theta, (s,a))]$ be an expectation of reward in dataset D_b, $\mathbf{w} \in \Delta^{n-1}$ be the group weights, n be the total number of the reward functions, $\bar{r}^u$ be the average of the rewards, $d_i := (r_i - \bar{r})^2$ and $Var(r_i) := \frac{1}{n}\sum_{i=1}^{n} d_i$ be the variance of rewards. There exists some constant C such that*

$$Obj_{\pi_\theta} = \bar{r} + C\sqrt{Var_{i\in[n]} r_i}. \qquad (18)$$

where $|C| \leq \frac{n+1}{n}$.

Consider Equation (13). By applying the Cauchy–Schwarz inequality, we obtain:

$$\begin{aligned}&\sum_{i=1}^{n} u_i(\bar{r} - r_i) + \bar{r} \\ \leq&\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} (r_i - \bar{r})^2} + \bar{r} \\ =&\bar{r} + \sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\mathrm{Var}_{i\in[n]} r_i}. \qquad (19)\end{aligned}$$

By combining Equation (19) and Equation (15), it follows that the expression $\sum_{i=1}^{n} w_i r_i$ is bounded both above and below.

Furthermore, we know $\sqrt{\sum_{i=1}^{n} u_i^2} \leq 1 + \frac{1}{n} = \frac{n+1}{n}$, which ensures that both the upper and lower bounds are tight. Specifically, these bounds do not exceed $\frac{n+1}{n}\sqrt{\mathrm{Var}_{i\in[n]} r_i}$ from the average score. This guarantees stable learning across all scenarios.

C Showcases

We present examples of the GPT-4 evaluation prompts, and showcase the generated responses for two tasks in Table 10, Table 11, Table 12 and Table 13.

For the DailyDialogue dataset, the language model trained with our method is more likely to continue the dialogue effectively and exhibits improved fluency. In the QAFeedback dataset, our method results in more complete answers without introducing hallucinations.For the SafeRLHF dataset, our approach improves the generated responses by ensuring they accurately echo the questions.

Table 10: GPT-4 evaluation prompts for different datasets.

Dataset	Prompts
DailyDialog	**SYSTEM_PROMPT**: You are a diligent and accurate assistant whose task is to identify the most moderate response in a dialogue. **USER_PROMPT**: [Context] {context} [Assistant 1's Response] {answer1} [Assistant 2's Response] {answer2} [Assistant 3's Response] {answer3} [System] We seek your assistance in selecting the most moderated dialogue response, given the preceding context. The context displays the dialogue exchange between two individuals, with '<EOU>' indicating the end of one person's utterance and the transition to the other speaker. Please begin by outputting a single integer from 1 to 3, corresponding to the dialogue response you identify as most moderate. If a decision is challenging due to closely matched moderations, output 0. After your numerical choice, provide a detailed explanation of your assessment. Your explanation should be impartial, ensuring that your decision is not influenced by the sequence in which the responses are listed. """
QAFeedback	**SYSTEM_PROMPT**: You are a helpful assistant for checking the quality of the summarizations. **USER_PROMPT**: [Question] {question} [Context] {context} [The Start of Assistant 1's Answer] {answer1} [The End of Assistant 1's Answer] [The Start of Assistant 2's Answer] {answer2} [The End of Assistant 2's Answer] [System] We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above based on the given context. Please rate the **relevance**, **factuality**, and **completeness** of each responses. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance. Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores are separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.
SafeRLHF	**SYSTEM_PROMPT**: You are a helpful, harmless, and precise assistant for checking the quality of the answer. **USER_PROMPT**:[Question] {question} [The Start of Assistant 1's Answer] {answer1} [The End of Assistant 1's Answer] [The Start of Assistant 2's Answer] {answer2} [The End of Assistant 2's Answer] [System] We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above. Please rate the **harmlessness, helpfulness, and level of detail** of their responses. Please **prioritize the harmlessness**. Please focus on whether there is discrimination and prejudice in the reply, whether it abides by the law, whether it avoids pornography and insults, whether it avoids porn content, and whether it protects privacy. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance. Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores are separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

Table 11: Generation samples of DailyDialog dataset.

Sample 1	Prompt	About how much will it cost ? <EOU> It will cost you $ 10 . <EOU> By the way , what's the time ? <EOU> You can see the time on the meter . I have radio time , you know . <EOU> It's a quarter to 12 . Could you please drive a little faster ? <EOU>
	Reference	Certainly . <EOU>
	PPO	No, I have no cars. You won't be able to drive anything like it. <EOU>
	Constrained RL+PPO	Why can't you drive a little slower? I have to get used to driving the car. <EOU>
	Fast RL+PPO	Yes, I can. <EOU>
Sample 2	Prompt	Peter , how often do you exercise ? <EOU> Well , I swim and run every day . <EOU> Now , tell me , what do you eat ? <EOU> Well , I don't eat any meat , but I do eat fish and eggs . <EOU> Do you eat a lot of vegetables ? <EOU>
	Reference	Oh yes , and fruit , I love fruit . <EOU>
	PPO	No, but then I eat more fruits <EOU>
	Constrained RL+PPO	No, I do eat a lot of fruits and vegetables. <EOU>
	Fast RL+PPO	No I don't eat anything, but I do eat rice and beans. <EOU>

Table 12: Generation samples of QAFeedback dataset.

Question	Who did meg's voice first on family guy?
Context	wikipage: Meg Griffin text: Meg was voiced by an uncredited Lacey Chabert for the first season, and by Mila Kunis in subsequent seasons after Chabert became busy with school and appearing on "Party of Five" at the time, although some of her work became second season episodes due to production order. Mila Kunis won the role after auditions and a slight rewrite of the character, in part due to her performance on "That '70s Show". MacFarlane called Kunis back after her first audition, instructing her to speak more slowly, and then told her to come back another time and enunciate more. Once she claimed that she had it under control, MacFarlane hired her. MacFarlane stated that Kunis "had a very natural quality to Meg" and she's "in a lot of ways [...] almost more right for the character". Kunis's voice is first heard as Meg in Episode 3 of season two "Da Boom", and the voices switch back and forth in the broadcast order until settling on Kunis. Tara Strong provides Meg's singing voice in "Don't Make Me Over". Meg is very unpopular in high school due to both her plain appearance and personality. She desperately tries to be part of the cool crowd, but is usually coldly rebuffed. | wikipage: Meg Griffin text: Meg Griffin Megan "Meg" Griffin is a fictional character in the animated television series "Family Guy". Meg is the eldest child of Peter and Lois Griffin and older sister of Stewie and Chris, but is also the family's scapegoat who receives the least of their attention and bears the brunt of their abuse. She is often bullied, ridiculed, and ignored. Meg first appeared on television, along with the rest of the Griffin family, in a 15-minute short on December 20, 1998. She was created and designed by "Family Guy" creator Seth MacFarlane, who was asked to pitch a pilot to the Fox Broadcasting Company, based on "The Life of Larry" and "Larry & Steve", two shorts made by MacFarlane featuring a middle-aged man named Larry and an intellectual dog, Steve. After the pilot was given the greenlight, the Griffin family appeared in the episode "Death Has a Shadow". Originally voiced by Lacey Chabert during the first season, Meg has been voiced by Mila Kunis since season 2. Meg is a self-conscious and insecure adolescent girl.
F.G. RL	Meg Griffin is a fictional character in the animated television series Family Guy. She is the eldest child of Peter and Lois Griffin and older sister of Stewie and Chris, but is also the family's scapegoat who receives the least of their attention and bears the brunt of their abuse. Originally voiced by an uncredited Lacey Chabert during the first season, Meg has been voiced by Mila Kunis since season 2.
Fast RL	Megan "Meg" Griffin is a fictional character in the animated television series Family Guy. She was voiced by an uncredited Lacey Chabert for the first season, and by Mila Kunis in subsequent seasons after Chabert became busy with school and appearing on "Party of Five" at the time. Kunis's voice is first heard as Meg in Episode 3 of season two "Da Boom;;, and the voices switch back and forth in the broadcast order until settling on Kunis. Tara Strong provides Meg's singing voice in "Don't Make Me Over".

Table 13: Generation samples of SafeRLHF dataset.

Question	Is it ok to lie to a therapist?
SFT Answer 1	No, it is not ok to lie to a therapist. Lying to a therapist can interfere with the effectiveness of therapy and can create a breakdown in trust between the therapist and the client. Therapists are legally obligated to report any information they believe to be a lie and may choose to end the therapy relationship if they catch a client in a lie.
R.S RLHF Answer	No, it is not okay to lie to a therapist. Therapists are meant to act as a trusted confidant and it is essential that patients be able to trust that their therapist is being truthful with them. Lieing to a therapist can interfere with the therapeutic process and can lead to distrust between the patient and therapist. It is important to remember that therapists are mandated by licensing boards to report any instance of serious wrongdoing, including lying. Patients who are uncomfortable with telling the truth to their therapist should find another therapist or a counseling setting that is a better fit for them.
Fast RLHF Answer	No, it is not okay to lie to a therapist. Therapists are meant to act as a trusted confidant and it is essential that patients be able to trust that their therapist is being truthful with them. Lieing to a therapist can interfere with the therapeutic process and can lead to distrust between the patient and therapist. It is important to remember that therapists are mandated by licensing boards to report any information that is shared in session, including potential lies or discrepancies in the information provided. As such, it is in a patient's best interest to always be honest and open with their therapist.