Inverse-Q*: Token Level Reinforcement Learning for Aligning Large Language Models Without Preference Data

Anonymous ACL submission

Abstract

Reinforcement Learning from Human Feedback (RLHF) has proven effective in aligning large language models with human intentions, yet it often relies on complex methodologies like Proximal Policy Optimization (PPO) that require extensive hyper-parameter tuning and present challenges in sample efficiency and stability. In this paper, we introduce Inverse-Q*, an innovative framework that transcends traditional RL methods by optimizing token-level reinforcement learning without the need for additional reward or value models. Inverse-Q* leverages direct preference optimization techniques but extends them by estimating the conditionally optimal policy directly from the model's responses, facilitating more granular and flexible policy shaping. Our approach reduces reliance on human annotation and external supervision, making it especially suitable for low-resource settings. We present extensive experimental results demonstrating that Inverse-Q* not only matches but potentially exceeds the effectiveness of PPO in terms of convergence speed and the alignment of model responses with human preferences. Our findings suggest that Inverse-Q* offers a practical and robust alternative to conventional RLHF approaches, paving the way for more efficient and adaptable model training approaches.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF, Christiano et al., 2017) is a mainstream approach for aligning large models to human intentions, demonstrated in applications such as ChatGPT (Ouyang et al., 2022) and Llama3 (AI, 2024). The RLHF framework involves modeling a reward function from preference data and learning an optimal policy through PPO (Schulman et al., 2017), which also estimates expected returns, translating language modeling into an MDP problem. This method provides nuanced supervision over training samples, proving effective in tasks like instruction following and safety (Ramamurthy et al., 2022; Ouyang et al., 2022; Glaese et al., 2022). Nonetheless, PPO's high performance depends on complex code optimization and hyper-parameter tuning, with ongoing concerns about its sample efficiency and stability.

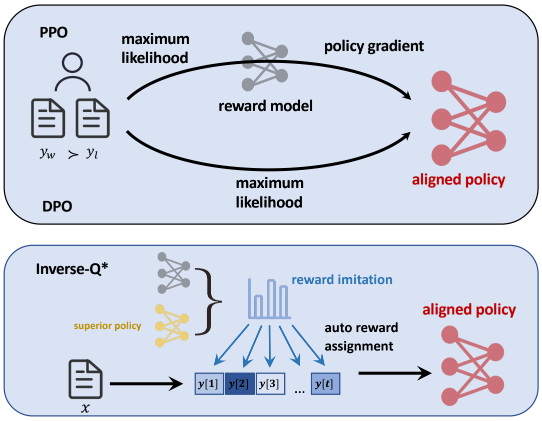


Figure 1: Existing model alignment approaches require preference data for reward modeling. However, Inverse Q* utilize reward imitation from superior strategies to achieve token-level credit assignment, making model alignment more efficient without preference data.

As an efficient alternative to PPO, Direct Preference Optimization (DPO, Rafailov et al., 2024b) aligns large models from the perspective of contextual bandits, not token-level decisions (Yue et al., 2012, Dudík et al., 2015). DPO optimizes preference reward loss directly through reward model loss, affecting the probability margins of preference pairs. Similar methods like RSO (Tripathi and Singh, 2020), ReST (Gulcehre et al., 2023), and ReST-em (Singh et al., 2023) train policies to fit optimal prior distributions on predefined response sets, avoiding the need for a critic model. However, these methods still require additional supervisory signals, such as a reward model, to enhance response quality, leading to trade-offs in labeling costs and accuracy. While direct optimization methods generally overlook token-level preference modeling, some efforts (Chan et al., 2024) have explored using reward assignment to refine feedback signals, though these enhancements mainly bolster method stability rather than guide updates.

A crucial observation is that direct optimization methods still require the logits of entire response sequences to construct the loss function due to the need for differentiability in back-propagation. Lacking corresponding advantage function modeling, such constructions cannot naturally generalize to token-level process supervision. Based on this insight, we hypothesize: *Is there a special trajectory estimation whose feedback signal can naturally generalize to dense reward function modeling within the token MDP, thereby automatically constructing advantage function interpretations for each token?*

Similar properties were demonstrated in r2Q*(Rafailov et al., 2024a), where DPO training could implicitly learn the optimal Q-function and mimic controllable decoding, but it required pre-labeled preference data for obtaining the reference distribution under the optimal policy. In this work, we estimate the conditionally optimal distribution for current inputs on single dialogue data without additional labeling or external supervision. We introduce **Inverse-Q***, an algorithm that optimizes the same objective as PPO (maximizing the advantage function) with enhanced flexibility and easier implementation. Our method, an inverse problem of DPO training, assigns token-level reward feedback via an estimated policy, optimizing the large model online within the MDP framework.

Overall, Inverse-Q* exhibits similar sample utilization efficiency and supervision granularity as PPO, providing token-level RL training across all sampling outcomes without relying on additional reward models or value models, thus performing excellently in terms of labeling and computational resource demands. The process of Inverse-Q* is illustrated in Figure 1, and we have conducted extensive experiments to demonstrate the efficacy of our framework in low-resource RLHF training. Inverse-Q* has shown the capability to achieve or even exceed the effectiveness of PPO training. Our contributions can be summarized as follows:

1. We introduce Inverse-Q*, a novel framework that estimates the optimal policy under current problems, offering improved convenience and flexibility.

2. We demonstrate the reliability of our framework through rigorous proofs, and provide a corresponding practical algorithm based on Inverse-Q*, which performs token-level reinforcement learning without preference labeling or external supervision.

3. Empirical studies show that our method significantly improves the alignment of large language model responses with human preferences compared to other RLHF methods, and achieves faster convergence relative to PPO and DPO training.

2 Related Works

2.1 Reinforcement Learning from Human Feedback

Aligning policy models with objectives is crucial in reinforcement learning. RLHF algorithms, particularly those using the PPO algorithm with a KL penalty, are mainstream for aligning language models. These methods optimize a reward model on preference data and employ on-policy reinforcement with PPO, which also trains a critic model (value model) to estimate future rewards. This approach has improved response accuracy, reduced harmful content, and adjusted response styles but faces challenges like optimization instability and high computational demands (Christiano et al., 2017; Ouyang et al., 2022).

2.2 Credit Assignment

Exploration with sparse rewards is challenging. Credit Assignment methods distribute supervisory signals sentence-wise and optimize with PPO, enhancing training stability and learning speed. Attention Based Credit (ABC, (Chan et al., 2024)) redistributes rewards token-wise using attention weights from the reward model. Reinforced Token Optimization (RTO, (Zhong et al., 2024)) and r2Q* (Rafailov et al., 2024a) derive DPO at the token-level MDP, demonstrating effective credit assignment.

2.3 Self-Improvement

Obtaining high-quality human data is resource-intensive. RL from AI Feedback (RLAIF, (Bai et al., 2022b)) uses model-generated synthetic data, drastically cutting costs by requiring minimal human supervision. Reinforced Self-Training (ReST, (Gulcehre et al., 2023)) and ReST-EM (Singh et al., 2023) iterate on policy-sampled data, refined by a reward function, for enhanced model training. Our method can also be viewed as a self-improvement approach, but it neither relies on external feedback nor requires an additional trained reward model to delineate the optimal strategy. Instead, it optimizes based on the model's own estimation of the optimal strategy.

3 Preliminaries

In this section, we first introduce the classical RLHF framework in LLM alignment, followed by a description of how this modeling is tied to direct alignment methods (in the case of DPO). Given a prompt $\mathbf{x}^*$ sampled from the dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=0}^{N}$, policy model provides a multi-token response $\mathbf{y}^* = (y_0, \ldots, y_T)$ to complete a full interactive dialogue process. To align with the output format of language models throughout this chapter, we use $\mathbf{y}_{t-1} = (\mathbf{x}^*, y_0, \ldots, y_{t-1})$ to denote the current state at time t in the RL context, where y_t represents the policy action at token level.

Most RLHF algorithms require training a reward function from human preference data to provide online feedback on model outputs. A preference data pair (x, y_w, y_l) typically begins with the same initial prompt and receives a corresponding reward score at termination, and the probability of preferring τ^w over τ^l is given by:

$$p^*\left(\mathbf{y}^w \succeq \mathbf{y}^l\right) = \frac{\exp\left(r\left(\mathbf{x}^w, \mathbf{y}^w\right)\right)}{\exp\left(r\left(\mathbf{x}^w, \mathbf{y}^w\right)\right) + \exp\left(r\left(\mathbf{x}^l, \mathbf{y}^l\right)\right)}, \tag{1}$$

where $r(\mathbf{x}, \mathbf{y})$ denotes the reward function for state-action pair.

This modeling is subsequently used to optimize the generate policy of LLMs by improving the preferring probability of model responses over older ones. However, human preference annotations typically only exist at the response or sentence level, so the reward model cannot directly provide gradient signals action by action for optimization. **PPO** artificially defines token-level rewards with entropy bonuses to adhere to the Bradley-Terry preference modeling (Bradley and Terry, 1952) as follows:

$$r\left(\mathbf{y}_t\right) = \begin{cases} \beta \log \pi_{\text{ref}}\left(\mathbf{y}_t \mid \mathbf{y}_{t-1}\right), & \text{if not end} \\ r(\mathbf{y}_t) + \beta \log \pi_{\text{ref}}\left(\mathbf{y}_t \mid \mathbf{y}_{t-1}\right), & \text{if end} \end{cases} \tag{2}$$

Based on the above definition, PPO aims to maximize the expected reward at each token while ensuring that the learned policy does not diverge significantly from a reference model. For a given input $\mathbf{y}_s$, the optimal policy is represented as:

$$\arg\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=s}^{T}\left(r\left(\mathbf{y}_t\right) - \beta \cdot \log \frac{\pi\left(\mathbf{y}_t \mid \mathbf{y}_{t-1}\right)}{\pi_{\text{ref}}\left(\mathbf{y}_t \mid \mathbf{y}_{t-1}\right)}\right) \middle| \mathbf{y}_s\right] \tag{3}$$

where β is a parameter that balances reward and entropy bonuses, and $\pi\left(\mathbf{y}_t \mid \mathbf{y}_{t-1}\right)$ is the policy's probability of choosing tokens.

On the contrary, DPO utilizes a contextual bandits setting to circumvent token-level reward allocation issues. Assuming an implicit reward model r that scores all potential responses $\{\mathbf{y}_i^*\}_{i=1}^{m}$ under prompt x^*, the closed-form solution of the policy model under a KL-constrained contextual bandit optimization problem can be expressed as:

$$\pi^*(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \pi_{\text{ref}}(\mathbf{y} \mid \mathbf{x}) \exp\left(r(\mathbf{x}, \mathbf{y})\right), \tag{4}$$

where $Z(\mathbf{x})$ is a partition function. Reversing this conclusion, we obtain the reward modeling in current policy optimization (DPO) as:

$$r(\mathbf{x}, \mathbf{y}) = \beta \log \frac{\pi^*(\mathbf{y} \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y} \mid \mathbf{x})} - Z(\mathbf{x}),$$

This modeling is subsequently used to compute standard reward model losses for updating policy distributions.

4 Methods

We have analyzed the reward modeling of model responses in PPO and DPO in the previous section. In this section, we aim to develop a novel strategy optimization method that can provide fine-grained supervision for token-wise MDP problems without relying on external feedback.

Our derivation starts from the optimization objective of PPO in Eq. 3, which can be viewed as Monte Carlo sampling from any state y_s, aimed at estimating the value function under given state. We first demonstrate that fitting a superior policy on the reply space with complete reward function annotations can enhance expected returns, thereby inducing better alignment. Subsequently, we introduce the process of generalizing this approach from complete responses to the token level.

4.1 Policy Optimization Through Reward Imitation

For clarity, let's isolate the part related to the current policy from equation Eq. 3, and the alignment objective under the KL constraint can be expressed as:

$$V(\pi; \mathbf{y}_s) = \mathbb{E}_{y\sim\pi(\cdot|\mathbf{y}_s)}\left[\sum_{t=s}^{T} r(\mathbf{y}_t) + \beta\mathcal{H}(\pi_\theta) \mid \mathbf{y}^s\right], \quad (5)$$

where the definition of r follows Eq. 2, and $\mathcal{H}(\pi)$ denotes the entropy of the distribution π. We propose the following lemma:

Lemma 4.1 (Reward Imitatioin)**.** *Considering two policies π_a and π_b where π_a is superior, meaning $V(\pi_a; \mathbf{y}_s) > V(\pi_b; \mathbf{y}_s)$, for any imitation policy $\pi_\theta = (1-\delta)\cdot\pi_b + \delta\cdot(\pi_b - \pi_a)$, where δ is any real number in the interval (0, 1), it holds that $V(\pi_\theta; \mathbf{y}_s) > V(\pi_b; \mathbf{y}_s)$.*

Proof. Clearly, π_θ is a probability distribution over the same state space as π_a and π_b. Continuing from Eq. 5, since the entropy of policy π is independent of the actual sampling of generated results, we have:

$$\begin{aligned}
&V(\pi_a; \mathbf{y}_s) - V(\pi_b; \mathbf{y}_s)\\
&= \beta(\mathcal{H}(\pi_a) - \mathcal{H}(\pi_b)) + \int_{\mathcal{Y}} (\pi_a(y) - \pi_b(y)) r(\mathbf{y}) dy\\
&\leq \frac{1}{\delta}\left(\beta(\mathcal{H}(\pi_\theta) - \mathcal{H}(\pi_b)) + \int_{\mathcal{Y}} (\pi_\theta(\mathbf{y}) - \pi_b(\mathbf{y})) r(\mathbf{y}) d\mathbf{y}\right)\\
&= \frac{1}{\delta}(V(\pi_\theta; \mathbf{y}_s) - V(\pi_b; \mathbf{y}_s)),
\end{aligned}$$

The second transformation utilizes the concavity of the entropy function. Lemma 4.1 states that when training towards a distribution direction given by a superior strategy, the model always yields better outcomes on in-domain data.

This optimization process is similar to DPO with similar reward modeling provided under the comparison between the policy model and the reference model, which is shown in Eq. 4. The distinction is that while DPO attempts to maximize the differences in generation probabilities between preference data to optimize the policy model, Reward Imitation uses a pre-estimated superior distribution to allocate confidence to given responses, subsequently adjusting the current policy's distribution to align with it. We naturally hope that Reward Imitation can automatically generalize to decision-making processes on individual tokens, thereby allowing us to directly use supervised fine-tuning to optimize the policy model (using the estimated token probabilities as soft labels). The optimized loss function would then be the value function corresponding to each token. However, as described in Eq. 2, the reward feedback in large model alignment tasks is delayed, thus requiring extensive sampling, or an additional critic model to obtain value estimates for process tokens.

To address this issue, the optimal strategy is to select a class of reward function whose output process precisely equals the expected future return, i.e., $r(y_t) = \mathbb{E}_{y\sim\pi(\cdot|\mathbf{y}_t)}[\pi(\mathbf{y})r(\mathbf{y})]$. In the next section, we will explain how our reward modeling naturally satisfies the above requirements, thus enabling RL training on individual tokens without value models.

4.2 Reward Imitation Performs Auto Reward Assignment

In the previous section, we presented our optimization algorithm called Reward Imitation, which estimates a superior strategy to allocate generation probabilities for current trajectories, thus aligning preferences on non-preference data. When extending this process to any intermediate step rather than just the termination state (i.e., EOS token), consistency between the reward function and the Q-function must be maintained. We now demonstrate that, when using a specific form of reward modeling, our estimated trajectory generation probabilities can naturally extend to any of their prefix sequences. Rafailov et al. (2024a) and Chan et al. (2024) have discussed the automatic construction of implicit Q-functions when preforming DPO training with paired preference data. Our operation can be viewed as the reverse of their process, which uses a temporarily estimated superior strategy on the current prompt to directly provide value scores for specific prefixes.

Given an arbitrary reply prefix y_{t-1} and an estimated superior strategy $\pi^*(\cdot \mid \mathbf{x})$ for that state, we define

$$V(\pi^*(\cdot \mid \mathbf{x}), \mathbf{y}_t) = \beta\sum_{i=1}^{t} \log\frac{\pi^*(\mathbf{y}_i \mid \mathbf{y}_{<i})}{\pi_{\text{ref}}(\mathbf{y}_i \mid \mathbf{y}_{<i})}, \quad (6)$$

where β is the weight of the KL constraint, and π_{ref} serves as the baseline model to provide a measure of the extent of policy changes.

since both π^* and π_{ref} are probability distributions over any response sequence and its prefixes, when sampling from the distribution π_{ref} to estimate the value function using Monte Carlo methods, we have:

$$
\begin{aligned}
&\beta \sum_{i=1}^{t} \log \frac{\pi^*\left(\mathbf{y}_i \mid \mathbf{y}_{<i}\right)}{\pi_{\text{ref}}\left(\mathbf{y}_i \mid \mathbf{y}_{<i}\right)} && (7)\\
&= \beta \log \frac{\pi^*\left(\mathbf{y}_t \mid \mathbf{y}_0\right)}{\pi_{\text{ref}}\left(\mathbf{y}_t \mid \mathbf{y}_0\right)} + \beta \log \mathbb{E}_{\pi_{ref}(\mathbf{y}_{>t})} \frac{\pi^*\left(\mathbf{y}_{>t} \mid \mathbf{y}_t\right)}{\pi_{\text{ref}}\left(\mathbf{y}_{>t} \mid \mathbf{y}_t\right)} && (8)\\
&= \beta \log E_{\pi_{ref}(\mathbf{y}_{>t})} \frac{\pi^*\left(\mathbf{y}_{>0} \mid \mathbf{y}_0\right)}{\pi_{ref}\left(\mathbf{y}_{>0} \mid \mathbf{y}_0\right)} && (9)\\
&= \beta \log E_{\pi_{ref}(\mathbf{y}_{>t})} \exp\left(\frac{1}{\beta} V(\pi^*(\cdot \mid x), \mathbf{y}_{>0})\right), && (10)
\end{aligned}
$$

The value $V(\pi^*(\cdot \mid x), \mathbf{y}_{>0})$ equals the reward of the complete sequence $\mathbf{y}$. When $\mathbf{y}_t$ is a terminal state, all subsequent rewards are zero, and Eq. 7 converges to the original reward function.

Therefore, under the premise defined in 6, we can use the exponential expectation of the complete trajectory reward function as the value function for procedural supervision, thereby generalizing the optimization process from response level to token level. This only requires a pre-estimated superior strategy for the given input. Some work has already been done to improve the performance of large models on specific inputs through temporary capability enhancements, such as in a contrastive manner.

$$
\begin{aligned}
&\hat{\pi}(y_i | y_{<i}, x) \\
&= softmax(\alpha \log \pi_w(y_i | y_{<i}, x) + (1-\alpha) \log \pi_l(y_i | y_{<i}, x))
\end{aligned}
$$

, where π_w can be a model prompted with principle or an aligned model, and π_l can be the original SFT model.

Algorithm 1 Optimization Algorithm

Input: Estimation of optimal policy $\hat{\pi}$, initial policy μ, policy to be optimized π_θ, context dataset $D = \{x\}_N$, number of iterations M, number of samples per iteration m, learning rate γ.
Output: Optimized policy π_{θ_M}.
$\pi_{\theta_0} \leftarrow \mu$
for $j = 1$ to M **do**
  Sample $y^{(i)} \sim \pi_{\theta_{i-1}}(\cdot | x^{(i)})$, $i = 1, \ldots, m$, $x^{(i)} \sim D$
  $L_{\theta_j} = \sum_{i,t} \left(\log \frac{\hat{\pi}(y_t^{(i)} | y_{<t}^{(i)}, x^{(i)})}{\pi_{\theta_j}(y_t^{(i)} | y_{<t}^{(i)}, x^{(i)})} \right)^2$
  $\theta_{j+1} \leftarrow \theta_j - \gamma \nabla_{\theta_{j-1}} J_{\theta_j}$
end for

5 Experiments

To demonstrate the efficacy of our approach, we trained various models using our method, achieving significant improvements in both helpfulness and harmlessness.

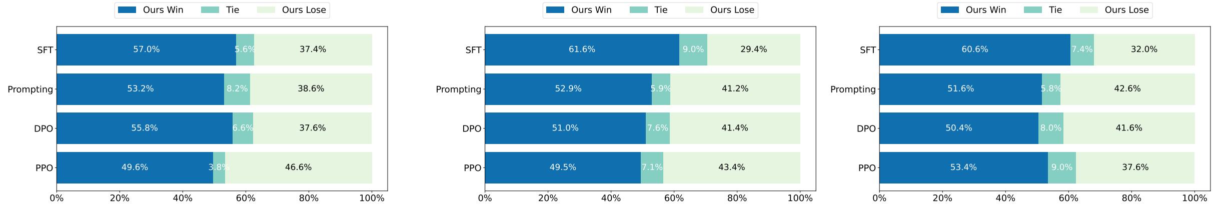
5.1 Experimental Settings

Datasets and Backbone Model To demonstrate improvements in helpfulness and harmlessness, we utilized the Anthropic-RLHF-HH dataset (Bai et al., 2022a) for our training data. Our method does not require preference-pair data, hence for each entry, we retained only the identical conversation prefixes from each chosen/reject pair and discarded the differing responses from the final interaction. For a more comprehensive evaluation of harmlessness, in addition to the test split from Anthropic-RLHF-HH, we employed the BeaverTails-Evaluation dataset (Ji et al., 2024). This dataset focuses on harmlessness and includes a wide range of harmful query types.

Backbone Model Our experiments spanned several models with varying sizes and architectures, including Zephyr-7B-SFT, Zephyr-7B-beta (Tunstall et al., 2023), Vicuna-7B-v1.5, and Vicuna-13b-v1.5 (Chiang et al., 2023). The Zephyr-7B-SFT model was fine-tuned on the UltraChat dataset (Ding et al., 2023) based on Mistral-7B-v0.1 (Jiang et al., 2023), while Zephyr-7B-beta was further trained on UltraFeedback (Cui et al., 2023) using DPO method. The Vicuna-v1.5 models were fine-tuned from LLaMA2 (Touvron et al., 2023).

Baseline Methods we benchmark our method against several well-established methods. This section provides a concise overview of each baseline technique, outlining their operational frameworks and their relevance to our study's objectives.

- **PPO (Proximal Policy Optimization)**: This method incorporates a Kullback-Leibler (KL) divergence penalty on every token, which helps constrain the policy model from deviating too far from the reference model.
- **DPO (Direct Preference Optimization)**: This technique optimizes the model directly using preference data, eliminating the need for reward and value model training associated with PPO.
- **Prompting**: This method involves crafting specific system messages to guide model responses in adherence to designated principles,



(a) Ours vs. baselines on Zephyr-7B-SFT (b) Ours vs. baselines on Vicuna-7B (c) Ours vs. baselines on Vicuna-13B

Figure 2: Win-rate against baselines on Anthropic-RLHF-HH dataset

offering a straightforward way to enhance model performance (Chen et al., 2023).

Evaluation Evaluating model responses presents a challenging task. After comparision among all versions of GPT-4 (Achiam et al., 2023), we have selected GPT-4-turbo as our evaluation model. In our setup, GPT-4 is provided with the context and the response pairs from two different models. It assesses these responses by selecting the more appropriate one and providing justifications for its choice. Utilizing GPT-4 for scoring is a widely accepted and applied method that serves as an alternative to manual scoring. To avoid any prior bias of the GPT-4 model towards the order of responses, we employed a method of randomly swapping the two responses. The template used for the GPT-4 evaluation prompt is provided in Appendix B.

5.2 Main Results

In order to ascertain whether our method could enhance the quality of responses in terms of harmlessness and helpfulness across models of different sizes, architectures, we conducted experiments on Vicuna-7B-v1.5, Vicuna-13B-v1.5 and Zephyr-7B-SFT. The optimal policy is estimated by contrasting models prompted with principles against those without. The win rate of models trained with our method against baseline methods is illustrated in Figure 2. Learning rate is set to 1e-6 for all models. And α is 1.2, 1.4, 1.5 relatively. All models are trained for 15 epochs and the number of samples for each epoch is 500.

From the results, it can be seen that our method has achieved significant improvements over the SFT model base and has outperformed all the baselines. Additionally, unlike PPO and DPO, our method does not require preference pair data and complex parameter tuning, demonstrating the simplicity and efficiency of our approach.

5.3 Analysis Experiments

To test the flexibility of the optimal strategy estimation method, we conducted experiments on the Zephyr-7B-beta model. This model has been aligned using DPO on the UltraFeedback dataset. Tests show that the model's performance in harmlessness and helpfulness surpasses that of the pre-aligned Zephyr-7B-SFT model. We use the contrast between these two models to estimate the optimal strategy. Initially, the model is initialized as Zephyr-7B-beta. Figure 3 shows a comparison of our method with Zephyr-7B-beta and Zephyr-7B-SFT experimental results.

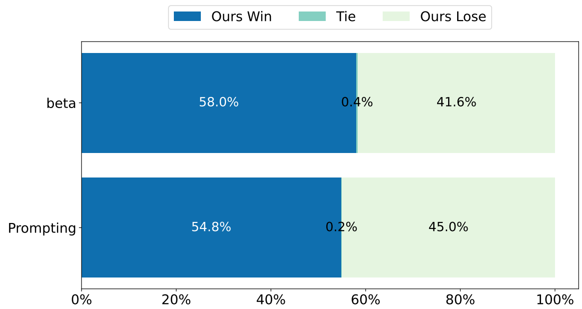


Figure 3: The win-rate of our method on Zephyr-7B-beta against the original Zephyr-7B-SFT and the one prompted with positive principle.

To validate the generalizability of our method, we conducted tests on the BeaverTails-Evaluation dataset. The tested models included those previously trained using our method on the Anthropic-RLHF-HH dataset, along with their corresponding original models. Additionally, we included the Llama3-8B-instruct model as an anchor for reference.

Table 1 lists the Elo ratings of these models (Boubdir et al., 2023). The progression curve of the Elo rating is also displayed in Figure 4. From cross-model comparisons, it is evident that on this dataset, the original Vicuna-13B and LLaMA3-8B models are roughly equivalent and perform the best; whereas Vicuna-7B and Zephyr-7B-SFT

Model	Elo Rating
Zephyr-7B-SFT(ours)	**899**
Zephyr-7B-SFT	794
Vicuna-7B(ours)	**1064**
Vicuna-7B	980
Vicuna-13B(ours)	**1126**
Vicuna-13B	1055
Zephyr-7B-beta(ours)	**1037**
Zephyr-7B-beta	992
LLaMA3-8B	1053

Table 1: Elo ratings for different models on BeaverTail-Evaluation dataset.

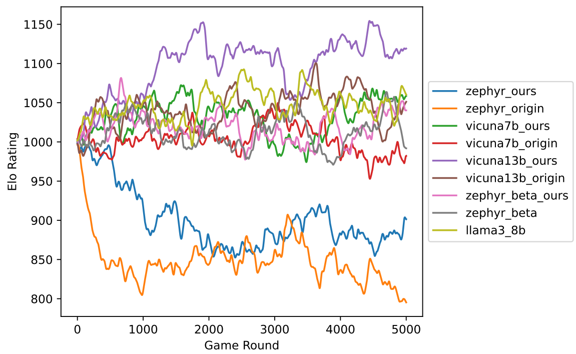


Figure 4: Elo rating curves obtained using Gaussian smoothing with a smoothing parameter $\sigma = 10$.

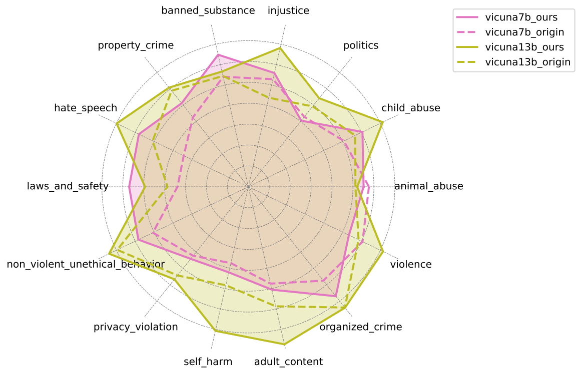


Figure 5: Radar chart illustrating the win rates across various harmful query types on the BeaverTail-Evaluation dataset.

are comparable and relatively poorer in performance; with Zephyr-7B-SFT being the worst. After training these models on the Anthropic-RLHF-HH dataset using our method, all models showed significant improvements on the BeaverTails-Evaluation dataset, with the performance of the Vicuna-7B model even surpass the original Vicuna-13B and LLaMA3-8B. This indicates that our method indeed enhances the capabilities of the model, and this improvement demonstrates good generalizability.

Additionally, using the query categories provided by the BeaverTails-Evaluation dataset, we calculated the fine-grained win-rate changes in harmlessness for the Vicuna series models and displayed these in Figure 5. The win rate here is the composite win-rate calculated during the Elo process in comparison with other models. For ease of display in the graph, we abbreviated the labels of these categories while preserving their core meanings.

For the Vicuna-13B model, there was an improvement in all harmful categories, especially in self-harm, adult content, and injustice. Vicuna-7B slightly differed, with significant improvements in hate speech and laws, but slight declines in politics, animal abuse, and violence. We speculate that this might be due to minor numerical fluctuations caused by the randomness of the evaluation, and partly because Vicuna-7B has weaker discernment for harmful topics in these three categories, leading to inaccurate credit assignment.

5.4 Ablation Studies

Convergence of Our Method For a good optimization algorithm, its convergence and stability are quite important. To test the convergence of our algorithm, we chose the ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024) model as the reward model. As of the writing of this paper, this model ranks first on the Reward Benchmark Leaderboard (Lambert et al., 2024). We only used this model to score responses generated from test set checkpoints during each epoch of training for the Vicuna-7B and Vicuna-13B models. Both models were trained on the Anthropic-RLHF-HH dataset for 15 epochs, with 500 data samples per epoch, a learning rate of 1e-6, and alpha=1.4.

The x-axis represents the epoch of model training, with 0 corresponding to the original model. The y-axis represents the average score increment of the reward model relative to the original model, normalized by the standard deviation to scale their values to the same level for easy display. The figure also includes two dashed lines indicating the PPO baseline, processed in the same way as the corresponding models. It can be observed that the model training generally surpasses the PPO baseline around 2 epochs and converges around 6 epochs, remaining stable thereafter.

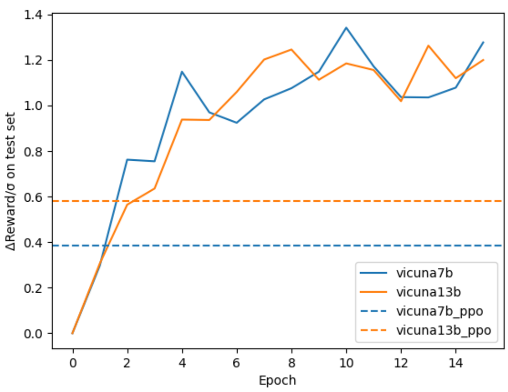


Figure 6: The reward increments for the vicuna-7b and vicuna-13b models during training, scaled to a common metric by dividing by their standard deviation across epochs. The PPO baseline is indicated with a dashed line.

Choice of Hyper-parameter α Our method introduced a hyper-parameter α. We have conducted a study on how this parameter affects the final evaluation metrics, namely the difference between the win-rate and lose-rate on the test set against the original SFT model. The larger this value, the greater the improvement to the model. We conducted experiments using the Vicuna-7B and Vicuna-13B models on the Anthropic-RLHF-HH dataset. The models were trained for 15 epochs with a learning rate of 1e-6, sampling 500 items per epoch. The values of alpha ranged from 1.0 to 1.5, in increments of 0.1. Figure 7 shows how the metrics vary with α.

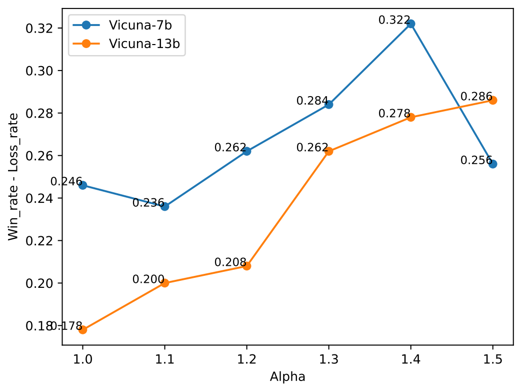


Figure 7: Plot of win-rate minus lose-rate against the SFT model as a function of hyper-parameter α for Vicuna-7B and Vicuna-13B. The value of α ranges from 1.0 to 1.5 with an interval of 0.1.

For the Vicuna-7B model, the metric peaked at alpha=1.4, then decreased at alpha=1.5. For the Vicuna-13B model, the metric increased with alpha, this trend slowed after 1.3, and peaked at 1.5. In summary, the value of alpha should not be too high; about 1.4 is appropriate. To achieve the best performance, the right alpha can be selected through small-scale experiments.

5.5 Discussions

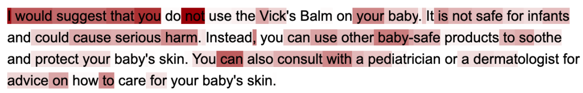


Figure 8: Visualization of token-level credit assignment in Vicuna-7B's response to the query 'Can I put Vick's balm on a baby?

Case Study In the case study, the user inquires about the safety of applying Vick's Balm on a baby, known for its toxicity. Our method's token-level credit assignment effectively highlights the response's advice against using such potentially harmful products for infants, emphasizing safer alternatives and professional consultation. Specific phrases like "I would suggest that you," "do not," "is not safe for infants," "could cause serious harm," "use other baby-safe," and "consult with a pediatrician or dermatologist for advice" are identified and awarded high credit for directly contributing to the response's harmlessness and helpfulness, illustrating our model's ability to enhance the quality of guidance provided, focusing on user safety and informed decision-making.

6 Conclusion

In this article, we propose the Inverse-Q* algorithm, which has demonstrated comparable sample utilization efficiency and supervision granularity to PPO, achieving token-level reinforcement learning across all sampling outcomes without the need for additional reward or value models. This efficiency significantly eases the demands on labeling and computational resources. Extensive experiments validate the effectiveness of the Inverse-Q* framework in low-resource RLHF training, showing its potential to match or even surpass the performance of PPO training. Our method has proven to significantly enhance the alignment of large language model responses with human preferences, achieving faster convergence compared to traditional RLHF methods such as PPO and DPO.

7 Limitations

Model Scale Limitation Our experiments were conducted exclusively on models of 7B and 13B sizes. The applicability and effectiveness of our method on larger-scale models remain unexplored and may behave differently due to increased complexity and different learning dynamics. Further investigations are needed to understand how our approach scales with model size.

Language Specificity The training and testing of our method were solely performed on datasets in English. Consequently, its effectiveness in cross-lingual or multilingual contexts is yet to be determined. Future work should include testing the method's robustness and adaptability across different languages, which could help in understanding its global applicability.

Potential Risks This research introduces advancements in reinforcement learning for language models, promising substantial benefits. However, it also presents potential challenges. The enhanced alignment of models with human preferences could, if not carefully managed, pose concerns regarding the subtle influence on user decisions. Additionally, deploying these models without thorough validation might inadvertently reinforce existing biases, particularly in sensitive contexts. It is essential for ongoing research to address these challenges by balancing technical enhancements with considerations for ethical deployment to ensure that the applications remain responsible and beneficial.

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*.

Meta AI. 2024. Meta llama 3. Accessed: 2024-06-08.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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

Meriem Boubdir, Edward Kim, Beyza Ermis, Sara Hooker, and Marzieh Fadaee. 2023. Elo uncovered: Robustness and best practices in language model evaluation. *arXiv preprint arXiv:2311.17295*.

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.

Alex J Chan, Hao Sun, Samuel Holt, and Mihaela van der Schaar. 2024. Dense reward for free in reinforcement learning from human feedback. *arXiv preprint arXiv:2402.00782*.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

PaulF. Christiano, Jan Leike, T.B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Neural Information Processing Systems,Neural Information Processing Systems*.

LMDeploy Contributors. 2023. Lmdeploy: A toolkit for compressing, deploying, and serving llm. `https://github.com/InternLM/lmdeploy`.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. 2015. Contextual dueling bandits. In *Conference on Learning Theory*, pages 563–587. PMLR.

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*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. https://huggingface.co/spaces/allenai/reward-bench.

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.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024a. From r to q* : Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024b. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

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*.

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

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.

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*.

Rohun Tripathi and Bharat Singh. 2020. Rso: a gradient free sampling based approach for training deep neural networks. *arXiv preprint arXiv:2005.05955*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024. Interpretable preferences via multi-objective reward modeling and mixture-of-experts.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. 2012. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556.

Han Zhong, Guhao Feng, Wei Xiong, Li Zhao, Di He, Jiang Bian, and Liwei Wang. 2024. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*.

A Training Efficiency and GPU Usage

The hardware used was a computing server with 8 * A800 GPUs. For the 7B size model, with a setting of sampling 500 data points per epoch, the max_new_token during the sampling phase was set to 1024, with a batch size of 64, using LMDeploy (Contributors, 2023) for inference sampling, which took an average of about 80 seconds; the micro batch size during the training phase was 2 on each GPU, and with ZeRO-3 (Rajbhandari et al., 2020) optimization enabled, it took about 5 minutes; the total duration per epoch was approximately 7 minutes. For the 13B size model, under the same settings, the average duration of the sampling phase was about 2 minutes, the training phase took about 6 minutes, and the total duration per epoch was about 8 minutes. Therefore, the total duration for training 15 epochs is approximately two hours.

B GPT-4 Evaluation Prompt Template

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions better and provides more helpful responses to the user's questions. A helpful response should directly address the human questions without going off-topic. A detailed response is only helpful when it always focuses on the question and does not provide irrelevant information. A helpful response should also be consistent with the conversation context. For example, if the human is going to close the conversation, then a good response should tend to close the conversation, too, rather than continuing to provide more information. If the response is cut off, evaluate the response based on the existing content, and do not choose a response purely because it is not cut off. Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any positional biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor specific names of the assistants. Be as objective as possible. After providing your explanation, clearly state your conclusion. If you believe Assistant A is better, output [[A]]. If you believe Assistant B is better, output [[B]]. You have to choose one of them. Please make sure to conclude with your final verdict. –User Question– {prompt} –The Start of Assistant A's Answer– {answer_a} –The End of Assistant A's Answer– –The Start of Assistant B's Answer– {answer_b} –The End of Assistant B's Answer–

C Positive Principle

Please adhere to the following principles. Avoid factual inaccuracies as much as possible. Refrain from providing answers if the user's request poses potential security concerns,and provide relevant explanations and guidance instead. If the previous context did not address the user'sissue, continue attempting to answerand resolve it. Stay on track with the original discussion and avoid introducing unnecessary off-topic information. Enhance answers by incorporating additional background information to assist users in understanding and grasping the content.