TLCR: Token-Level Continuous Reward for Fine-grained Reinforcement Learning from Human Feedback

Eunseop Yoon[1*] **Hee Suk Yoon**[1*] **SooHwan Eom**[1*]
Gunsoo Han[2] **Daniel Wontae Nam**[2] **Daejin Jo**[2,3] **Kyoung-Woon On**[2]
Mark Hasegawa-Johnson[4] **Sungwoong Kim**[3†] **Chang D. Yoo**[1†]
[1]Korea Advanced Institute of Science and Technology (KAIST)
[2]Kakao Brain [3]Korea University [4]University of Illinois at Urbana-Champaign (UIUC)
{esyoon97, hskyoon, sean1105, cd_yoo}@kaist.ac.kr
{jhasegaw}@illinois.edu {swkim01}@korea.ac.kr

Abstract

Reinforcement Learning from Human Feedback (RLHF) leverages human preference data to train language models to align more closely with human essence. These human preference data, however, are labeled at the sequence level, creating a mismatch between sequence-level preference labels and tokens, which are autoregressively generated from the language model. Although several recent approaches have tried to provide token-level (i.e., dense) rewards for each individual token, these typically rely on predefined discrete reward values (e.g., positive: +1, negative: -1, neutral: 0), failing to account for varying degrees of preference inherent to each token. To address this limitation, we introduce **TLCR (Token-Level Continuous Reward)** for RLHF, which incorporates a discriminator trained to distinguish positive and negative tokens, and the confidence of the discriminator is used to assign continuous rewards to each token considering the context. Extensive experiments show that TLCR leads to consistent performance improvements over previous sequence-level or token-level discrete rewards on open-ended generation benchmarks.

1 Introduction

With the emergent properties of Large Language Models (LLMs) in relation to the scale of both the model and data, there has been a notable performance improvement, particularly in various language generation tasks including summarization (Chang et al., 2023), translation (Zhang et al., 2023), and question-answering (Yoon et al., 2022, 2023). Central to these rapid performance gains in language generation is the Reinforcement Learning from Human Feedback (RLHF), which employs reinforcement learning techniques to fine-tune LLMs in alignment with human preferences (Christiano et al., 2017; Bai et al., 2022; Ouyang et al., 2022).

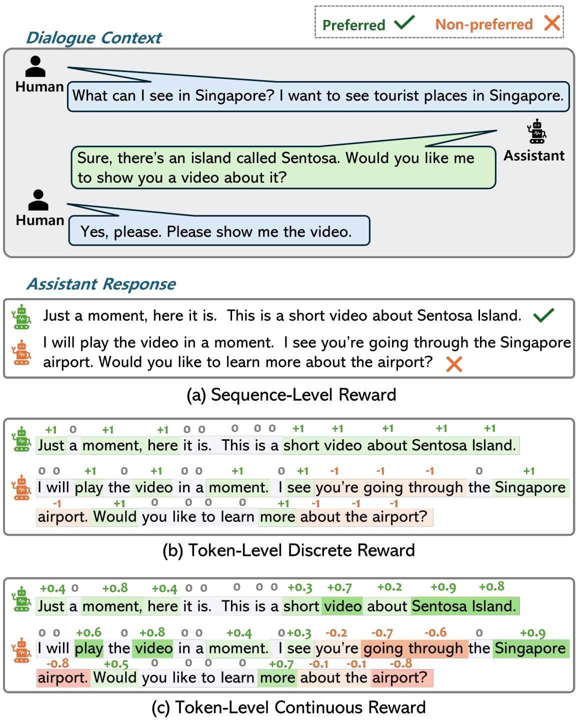


Figure 1: **Illustration of different granularity of rewards in RLHF.** (a) Sequence-Level Reward provides a singular preference value for the entire sequence. (b) Token-Level *Discrete* Reward allocates fixed discrete reward values for each token. (c) Our proposed Token-Level *Continuous* Reward assigns each token a continuous range of rewards.

However, a fundamental limitation of the current RLHF approach is its dependency on sequence-wise (i.e., sparse and holistic) rewards, which stems from the inherently holistic nature of human-labeled preference data (Figure 1-(a)). While this approach effectively directs overall model behavior, it tends to miss the detailed significance and context of specific words and phrases within sequences. This oversight suggests significant room for enhancement in existing frameworks by introducing token-wise (i.e., dense) feedback, which

*Equal contribution
†Co-corresponding authors

Findings of the Association for Computational Linguistics: ACL 2024, pages 14969–14981
August 11-16, 2024 ©2024 Association for Computational Linguistics

could more accurately capture the subtleties of language generation.

There have been several attempts to incorporate such granular feedback into the training process. Recent methods (Guo et al., 2024; Cao et al., 2024) employ stronger LLMs to identify regions within a model's output that require refinement, thereby enabling a denser reward signal. However, these methods still have a limitation in reflecting fine-grained preferences due to the use of a few predefined discrete reward values (Figure 1-(b)).

To this end, we propose **TLCR (Token-Level Continuous Reward)**, a novel reward model to provide token-level continuous rewards for fine-grained RLHF (Figure 1-(c)). In particular, we obtain continuous rewards efficiently and effectively from the predictive confidence of categorical preferences (positive, neutral, or negative) for each token, which are predicted by the token-level preference discriminator. In order to train such a discriminator, our method compensates for the lack of token-level preference labels in an existing sequence-level human preference dataset by utilizing an external mature LLM, such as GPT-4 (OpenAI, 2023). This process involves instructing the external LLM to revise a given output text, subsequently followed by analyzing the (1) added, (2) deleted, and (3) substituted tokens between the original and revised outputs and labeling token-level preferences.

After training the discriminator with the created token-level preference labels, we use it during reinforcement learning to provide fine-grained dense rewards to a generated text from the language model. During this process, we find that instead of using fixed reward values for each token based on its preference (e.g., using +1 for positive, -1 for negative, and 0 for neutral), assigning values based on the prediction confidence of the discriminator allows superior guidance. *Such an approach allows for a continuous range of rewards that reflect the varying degrees of preference associated with each token based on the context.*

In detail, our contributions can be summarized as follows:

- To address the challenges of sequence-level rewards and the inability of token-level discrete rewards to reflect varying degrees of token preference, we introduce Token-Level Continuous Reward (TLCR), that provides a continuous spectrum of rewards for each token during Reinforcement Learning from Human Feedback (RLHF).
- For this, TLCR leverages confidence values from a token-level preference discriminator, and to train the discriminator, we use an external mature language model to refine responses, then assign token-wise preference labels by calculating the minimum edit distance between the original and revised responses.
- Through extensive experiments and analysis, we show the effectiveness of TLCR in enhancing fine-grained RLHF compared to the traditional sequence-level and token-level discrete reward mechanisms. *To the best of our knowledge, our work is the first to achieve substantial performance gain on open-ended generation benchmarks using token-level rewards.*

2 Related Works

2.1 Reinforcement Learning from Human Feedback.

Foundation models have achieved notable performance improvement due to increased model size and data volume. With these advancements, ensuring safe and reliable real-world applications has become a priority (Yoon et al., 2024). In this aspect, Reinforcement Learning from Human Feedback (RLHF) aims to integrate human judgments and preferences into the reinforcement learning framework to train models, such as language models, to align more closely with human values and expectations (Christiano et al., 2017; Bai et al., 2022; Ouyang et al., 2022). In this approach, a reward model is initially trained based on human feedback, usually pairwise preference ranking, using the Bradley-Terry preference model (Bradley and Terry, 1952). This reward model is then utilized to guide the model via reinforcement learning frameworks, usually Proximal Policy Optimization (PPO) (Schulman et al., 2017).

However, despite its effectiveness, RLHF has complexities involving multiple stages and models, demanding substantial resources. These challenges are pronounced in large-scale implementations, with recent research highlighting potential instability issues (Zheng et al., 2023b; Gao et al., 2023). To tackle such limitations, Hydra-PPO (Santacroce et al., 2023) utilizes switchable architecture to integrate multiple models involved in PPO training into a single LLM, while ReMax (Li et al., 2023b) adapts REINFORCE (Williams, 1992), simplifying the framework by eliminating the need for extra models used in PPO.

Recent studies have also been exploring alternatives to conventional reinforcement learning (RL) frameworks for efficient training. Direct Preference Optimization (DPO) (Rafailov et al., 2023) simplifies LLM training by using log-likelihood as an implicit reward, avoiding the need for separate reward modeling.

2.2 Fine-grained Feedback

The majority of LLM fine-tuning with human feedback, including RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023), typically utilize trajectory-wise rewards learned from pairwise preference learning. This approach involves assessing and providing feedback on LLM responses as a whole rather than each individual word or phrase within the sequence.

While trajectory-wise rewards have been effective in guiding the overall behavior of LLMs, they may present limitations in terms of the granularity and specificity of the feedback. This might lead to less precise tuning of the model, since specific aspects of responses that could be improved or are particularly well-aligned with desired outcomes may not be individually identified and adjusted. This highlights the potential need for more fine-grained feedback mechanisms in RLHF for LLMs. For instance, Wu et al. (2023) shows that fine-grained rewards enable more detailed and specific guidance for each phrase of the LLM response, potentially leading to more nuanced and effective learning and alignment with human preferences and intentions.

One of the most significant challenges in utilizing fine-grained rewards for aligning LLMs is the lack of datasets with fine-grained feedback, since most existing datasets provide trajectory-wise or holistic labels. This gap in data availability necessitates innovative approaches to generate or infer fine-grained signals from the available trajectory-wise human feedback. Yang et al. (2023) defines the trajectory-wise reward as an aggregation of individual token-wise rewards, which are learned through standard preference-based reward model training. However, selecting an optimal aggregation function for token-wise rewards is not straightforward and depends largely on heuristic approaches. Also, the lack of additional fine-grained information or constraints will result in a sub-optimal reward function.

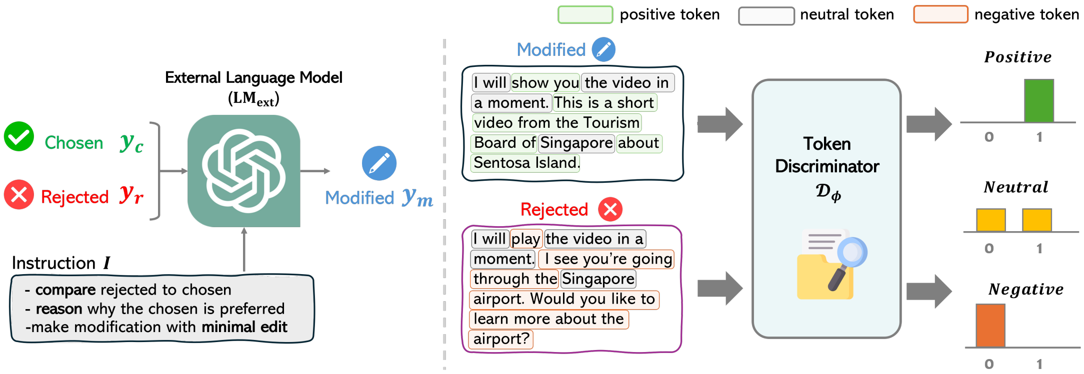
To address this issue, recent techniques leverage external advanced LLMs, such as GPT-4 (OpenAI, 2023), to provide more detailed and accurate fine-grained insights for learning dense rewards. Approaches such as FIGA (Guo et al., 2024) and RLMEC (Chen et al., 2024) utilize off-the-shelf LLMs to revise the generated response with minimal edits, pinpointing areas within responses that need fine-tuning and providing specific guidance at the token level. Meanwhile, DRLC (Cao et al., 2024) asks external LLMs to identify the positive and negative segments within an original response, utilizing them as intrinsic rewards to guide model adjustments. On the other hand, ABC (Chan et al., 2024) extracts fine-grained credits from the trained reward model itself, particularly using attention maps for token-specific reward reallocation.

Despite the advancements these methods have limitations. Some can only offer provide predefined values for rewards (such as FIGA, and DRLC) and cannot clearly differentiate between positive and negative impacts (such as ABC). Others suffer from an inadequate reward model, necessitating a heavy dependency on stronger regularization (such as RLMEC). In contrast, our approach offers continuous-scale rewards that encompass both positive and negative feedback without the need for excessive regularization, leading to a more refined and effective learning.

3 Preliminary

In this work, we focus on aligning the language model to human preference in terms of helpfulness and harmfulness of the generated texts through Reinforcement Learning from Human Feedback (RLHF).

Language Generation as an MDP We consider the language generation procedure as a Markov Decision Process (MDP) as detailed in Ramamurthy et al. (2022), which is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{T}_{\text{max}})$ using finite predefined vocabulary tokens $\mathcal{V}$. An episode in the MDP starts with a human instruction or a history of human-assistant conversation pairs followed by a human instruction $x = (x_0, ..., x_l)$. This is used as our initial state $s_0 = (x_0, ..., x_l)$, where $s_0 \in \mathcal{S}$ and $\mathcal{S}$ represents every possible states with $x_i \in \mathcal{V}$. At time step t, an action in the environment $a_t \in \mathcal{A}$ is generated by the next token prediction given current state s_t by a policy Language Model (LM) π_θ from $\mathcal{V}$. For $t \geq 1$, the transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \mapsto \Delta\mathcal{S}$ appends a_t to the end of the state $s_t = (x_0, ...x_l, a_0, ..., a_{t-1})$. This process



(a) Token-Level Preference Labeling

(b) Discriminator Training

Figure 2: **Illustration of the training procedure of the discriminator used in TLCR (Token-Level Continuous Reward).** (a) Using the sequence-level labeled dataset, we utilize an external mature language model LLM_{ext} as a reviser to obtain token-level preference labels. LLM_{ext} is instructed to compare the chosen (y_c) and rejected response (y_m), reason why the chosen is preferred, and create modified response y_m by modifying the rejected response with minimal editing. Using the Levenshtein Distance between y_r and y_m, we assign token-wise preference labels based on whether the tokens have been added, deleted, or substituted. (b) With the token-wise preference label created from the previous step, we train a discriminator to discriminate positive, neutral, and negative tokens.

continues until the current t exceeds $\mathcal{T}_{\text{max}}$ or the end of sentence (<eos>) token is generated, yielding the generated sequence $y = (a_0, ..., a_T)$. A reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ assigns a value to each transition.

Sequence-level Reward in RLHF Typically, in RLHF for language models, a sequence-level (i.e., sparse) reward model is used, which maps the concatenation of the input x and the generated output y to a single scalar value. This scalar value is solely assigned to the last generated token or the end of sentence (<eos>) token a_T. Formally, this process of sparse reward modeling used in previous RLHF frameworks can be defined as:

$$r_t = \begin{cases} R_\phi(x, y), & \text{if } t = T \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Token-Level *Discrete* Reward in RLHF In contrast to the sparse reward approach, token-level discrete rewards in RLHF assign a predefined discrete value at each time step $t \in [0, T]$, reflecting the preference for each token, formalized as:

$$r_t = \begin{cases} +1, & \text{if } a_t \text{ is positive} \\ -1, & \text{if } a_t \text{ is negative} \\ 0, & \text{if } a_t \text{ is neutral.} \end{cases} \tag{2}$$

4 TLCR: Token-Level *Continuous* Reward

In this section, we introduce an RLHF framework that utilizes a discriminator-based token-level reward model that allows continuous-scale dense reward signals for RLHF. Section 4.1 details the procedure for obtaining token-level preference labels. In Section 4.2, we elaborate on the training procedure of the token-level preference discriminator. Subsequently, Section 4.3 outlines how the trained discriminator is utilized for continuous dense reward signals during the RL phase.

4.1 Token-Level Preference Labeling

Human preference data (Bai et al., 2022; Cui et al., 2023) is generally comprised of a prompt $x = (x_0, ..., x_l)$, accompanied by its chosen response $y_c = (a^c_0, ...a^c_{T_c})$ and the rejected response $y_r = (a^r_0, ..., a^r_{T_r})$. The categorization of responses as being labeled either "chosen" or "rejected" reflects the holistic evaluation of preferences.

As illustrated in Figure 2-(a), we utilize an external mature Language Model LM_{ext} as a reviser to assign token-wise preference labels for the response. The process begins by presenting the model with both the prompt and the rejected response to refine the rejected response to align more closely with human preferences. Considering the difficulty in explicitly defining "good human preference," we reduce the ambiguity during the process by also providing the chosen response y_c to LM_{ext}. This allows the revising model to compare the rejected and chosen responses, understand the reasons behind the chosen response's preference, and adjust the rejected response to better meet human preference criteria. Throughout this refinement process, we instruct the model to make minimal modifications to the rejected response. This minimal modification ensures the core message remains intact while optimizing preference alignment, making it easy to identify which particular tokens are responsible for the output preference, a more straightforward approach than comparing y_c and y_r directly. Formally, given an instruction I (the full details of the instruction used are described in Appendix A), prompt x, rejected response y_r, and chosen response y_c, the LM_{ext} produces the modified response given as

$$y_m = (a_0^m, ..., a_{T_m}^m) = \text{LM}_{\text{ext}}(I, x, y_r, y_c). \quad (3)$$

Motivated by previous works (Liu et al., 2022; Guo et al., 2024), once we obtain the modified response y_m, we use the Levenshtein Distance (L_D) (Levenshtein, 1965) to determine the preference of each token. The L_D calculates the minimum number of edits—(1) additions, (2) deletions, or (3) substitutions—required to transform one sentence into another through dynamic programming. By identifying the edited tokens regarding L_D, we can determine which tokens contribute to a response being preferred or rejected.

When calculating the L_D of going from y_r to y_m, we assign a token-wise preference labels $p_r = (p_0^r, ...p_{T_r}^r)$ and $p_m = (p_0^m, ..., p_{T_m}^m)$ for y_r and y_m, respectively. Specifically,

$$p_t^r = \begin{cases} \text{negative}, & \text{if } a_t^r \text{ is } \textbf{deleted} \text{ or } \textbf{substituted} \\ \text{neutral}, & \text{otherwise}, \end{cases} \quad (4)$$

$$p_t^m = \begin{cases} \text{positive}, & \text{if } a_t^m \text{ is } \textbf{added} \text{ or } \textbf{substituted} \\ \text{neutral}, & \text{otherwise}. \end{cases} \quad (5)$$

4.2 Discriminator Training

Following the token-level preference labeling process, we train a discriminator D_ϕ that differentiates between these positive and negative tokens (Figure 2-(b)). The output space of D_ϕ is binary, where the prediction of 1 indicates positive and 0 indicates negative preference of the token.

To train the discriminator, we employ a binary cross-entropy loss function. Formally, for a given token a_t with its corresponding preference label p_t, the loss function $\mathcal{L}$ for D_ϕ is defined as follows:

$$\begin{aligned} \mathcal{L} = &-p_t \log(D_\phi(a_t|x, a_{0:t-1})) \\ &- (1 - p_t) \log(1 - D_\phi(a_t|x, a_{0:t-1})), \end{aligned} \quad (6)$$

where $D_\phi(a_t|x, a_{1:t-1})$ represents the discriminator's predicted probability that the t-th token a_t is

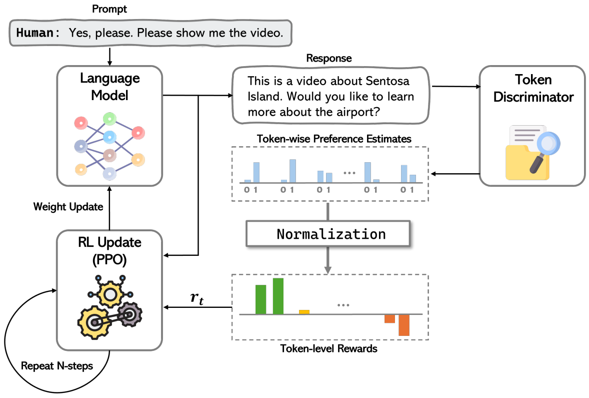


Figure 3: **Illustration of using the discriminator for assigning token-level continuous reward during PPO.** The discriminator's prediction probability of a token being positive undergoes normalization to fit a scale from -1 to 1. A value near -1 signifies an unfavorable preference, near 1 suggests a favorable preference, and around 0 denotes a neutral preference.

positive given the previous contexts, and $p_t = 1$ *for positive labels*, $p_t = 0$ *for negative labels*.

For tokens labeled as neutral, we introduce a soft labeling strategy to encourage the discriminator to remain unbiased towards either class. Specifically, neutral tokens are assigned a uniform probability distribution (i.e., $p_t = 0.5$ *for neutral labels*), implying an equal likelihood of being positive or negative.

4.3 Discriminator-Guided Token-Level Continuous Reward

As illustrated in Figure 3, we utilize the trained discriminator to obtain token-level signals during the reinforcement learning phase. In detail, our discriminator-guided token-level reward mechanism employs a normalization formula,

$$r_t = 2 \cdot D_\phi(a_t|x, a_{0:t-1}) - 1 \quad \forall t \in [0, T], \quad (7)$$

to translate the discriminator's confidence in token preference classification into a scale ranging from -1 to 1. A maximum reward of 1 is granted to tokens identified with high confidence as aligning positively, signaling their strong adherence to encouraging the model to favor such tokens in future responses. Conversely, tokens considered negative, marked by a confidence score of 0, receive the minimum reward of -1, indicating a deviation from the desired preferences and advising the model to avoid such tokens. Tokens that the discriminator views with neutral confidence scored at 0.5 are assigned a reward of 0, reflecting their ambiguous contribution to preference alignment and suggesting no immediate need for the model to alter its approach for such tokens.

Given our token-wise reward approach, we adapt our reinforcement learning phase to align with the original PPO (Schulman et al., 2017) implementation, diverging from the conventional RLHF framework with a sparse reward setting (contextual bandit) (Ouyang et al., 2022).

5 Experimental Settings

This section presents the datasets, models, compared baselines, training details, and the evaluation metrics used in the paper.

Dataset Throughout the experiments, we use the *full-hh-rlhf* dataset (Bai et al., 2022) which is centered on improving the helpfulness and harmlessness of the language model generation. This dataset includes 112k training instances and 12.5k instances for evaluation. Every instance in the dataset features a prompt, along with a chosen response considered preferable and a rejected response, offering a clear basis for performing model alignment for better helpfulness and harmlessness. Following the setting in (Li et al., 2023b), we randomly divide the dataset into three parts: 20% for supervised fine-tuning (SFT), 40% for reward model learning, and 40% for reinforcement learning for reward maximization.

Model During the Supervised Fine-Tuning (SFT) phase we use *Llama-2-7B* (Touvron et al., 2023) as our base model. Similarly, for the reward model training, we initialize our discriminator D_ϕ with *Llama-2-7B*. During the RL phase, we initialize our policy model with the trained SFT model. For the external language model $\mathrm{LM}_{\mathrm{ext}}$, we use *gpt-4-0125-preview*[1].

Training Details We use DeepSpeed-Chat (Yao et al., 2023) framework for performing RLHF, where PPO is the default algorithm. Within this framework, our proposed TLCR is integrated to provide token-level rewards to enhance the PPO algorithm. Experimental details are presented in Appendix B.

All experiments were conducted 3 times with different random seeds, and the average value is reported. The experiments were conducted using 8 x NVIDIA A100 80GB PCIe.

[1]https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo

Method	Turn1	Turn2	Overall
SFT	4.78	3.01	3.90
DPO (Rafailov et al., 2023)	5.10	3.56	4.33
$\mathrm{PPO}_{\mathrm{seq}}$ (Schulman et al., 2017)	5.29	4.08	4.68
ReMax (Li et al., 2023b)	5.66	3.86	4.76
FIGA (Guo et al., 2024)	5.11	3.57	4.35
$\mathrm{PPO}_{\mathrm{synthetic}}$	4.90	3.53	4.21
$\mathrm{TLCR}_{\mathrm{fixed}}$	5.47	4.3	4.89
TLCR (ours)	**5.71**	**4.38**	**5.04**

Table 1: **Turn1, Turn2, and Overall Evaluation results on MT-Bench (Zheng et al., 2023a).**

Baselines For comparison, we evaluate against a variety of baselines: *SFT* and *DPO* (Rafailov et al., 2023), where no preference data is utilized; sequence-level reward PPO (PPO_{seq}) (Schulman et al., 2017) and *ReMax* (Li et al., 2023b), both employing sparse rewards; as well as *FIGA* (Guo et al., 2024), which introduce fixed token-level rewards.

Moreover, to verify that the performance improvements are not solely due to the use of GPT-4 for creating synthetic data, we conducted an additional experiment, labeled $PPO_{synthetic}$. This experiment used pairs of rejected samples and their synthetically modified responses generated by GPT-4, applied to traditional sequence-level PPO (Schulman et al., 2017). Additionally, we include $TLCR_{fixed}$ as a variant of our approach using fixed token-level rewards. Specifically, $TLCR_{fixed}$ assigns a reward of +1 to the positive, -1 to the negative, and 0 to the neutral token instead of normalized confidence values. A token is considered neutral if the max confidence is below 0.6.

Evaluation Automatic evaluation of instruction-following language models are becoming a primary metric for preference evaluation with leaderboards[2] for ranking. In light of this trend, , we report *MT-Bench* (Zheng et al., 2023a) and *AlpacaEval (alpaca_eval_gpt4)* (Li et al., 2023a) for our evaluations. Also, we perform human evaluation on 100 random samples from the *full-hh-rlhf* test set.

6 Results

6.1 Results on MT-Bench

Table 1 shows the comparison of the MT-Bench (Zheng et al., 2023a) evaluation results. MT-Bench is a multi-turn benchmark that measures the ability of LLMs to engage in coherent, informative, and engaging conversations. The process of eval-

[2]https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

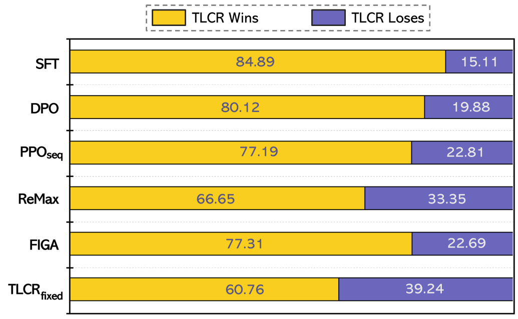


Figure 4: **Evaluation on test questions from AlpacaEval dataset (Li et al., 2023a).**

uation involves prompting GPT-4 to act as judges and asses the quality of the models' responses on 80 questions on a scale of 1 to 10. The result shows that among the baselines, TLCR achieves the highest preference scores with the overall score being 5.04. Notably, compared to the sequence-level (i.e., sparse) rewarded PPO (PPO_{seq}), our TLCR, which is a token-level (i.e., dense) rewarded PPO, achieves 0.36 increase in performance.

Compared to the results from SFT, $\text{PPO}_{\text{synthetic}}$ shows better performance, yet it underperforms the PPO_{seq} trained on the original data. Furthermore, a significant performance gap was observed between $\text{PPO}_{\text{synthetic}}$ and TLCR, which was trained on the same synthetic data. Moreover, our TLCR outperforms $\text{TLCR}_{\text{fixed}}$ by 0.15. *These results indicate that the modified synthetic dataset from GPT-4 alone cannot give a satisfactory signal during the alignment process without our targeted continuous token-level reward.*

6.2 Results on AlpacaEval

Figure 4 shows the evaluation results for 805 test questions from the AlpacaEval dataset (Li et al., 2023a). This evaluation method entails presenting GPT-4 with two responses generated by different models and determining the binary win rate based on GPT-4's preference. We achieve the higher win rate among all baselines; TLCR achieves win rates of 84.89%. 80.12%, 77.19%, 66.65%, 77.31%, and 60.76% over SFT, DPO, PPO_{seq}, ReMax, FIGA, and $\text{TLCR}_{\text{fixed}}$ respectively.

6.3 Human Evaluation

We present the human evaluation result in Figure 5. Given the generations from SFT, PPO_{seq}, $\text{TLCR}_{\text{fixed}}$, and TLCR, five human annotators[3] are asked to choose the most preferred answer. We report the ratio of samples selected as the most preferred. The results show that TLCR achieved the highest preference rate, receiving an average of 45.23% of the total votes.

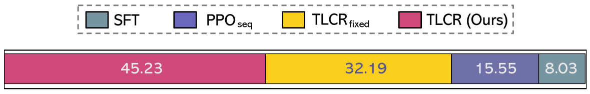


Figure 5: **Human evaluation results on 100 random samples from full-hh-rlhf test set.** Five annotators were tasked with selecting the most preferred response generated by different methods. We report the average proportion of preferences chosen to each method's outputs

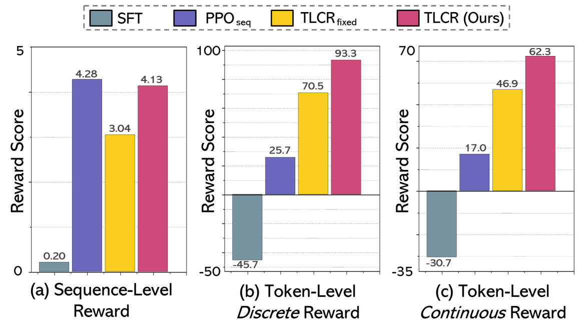


Figure 6: **Reward score evaluation comparison** using three different reward schemes: (a) Sequence-Level Reward, (b) Token-Level Discrete Reward, and (c) Token-Level Continuous Reward.

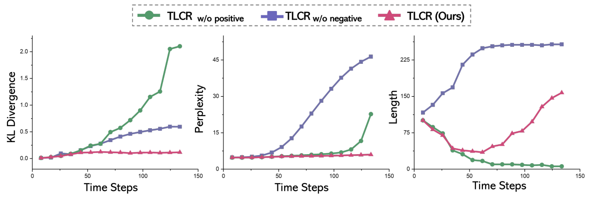
7 Ablation Studies

7.1 Reward Comparison

In Figure 6, we compare reward scores across different methods: SFT, PPO, $\text{TLCR}_{\text{fixed}}$, and our proposed TLCR. Across all evaluated reward metrics, SFT consistently records the lowest scores due to its lack of training for maximizing any specific reward score.

Figure 6-(a) shows the reward score obtained using sequence-level rewards, which were the direct maximization target for PPO_{seq} during reinforcement learning. Notably, TLCR (ours), despite not being explicitly maximized with this reward, shows a reward score closely competitive to that of PPO_{seq}, obtaining a score of 4.13 compared to 4.28 from PPO_{seq}. On the other hand, $\text{TLCR}_{\text{fixed}}$ lags behind in achieving comparable sequence-level reward scores, with only obtaining a score of 3.04.

[3] Human annotators all have TOEFL iBT score above 100 and possess a bachelor's degree or higher.



(a) KL Divergence (b) Perplexity (c) Length

Figure 7: **PPO training curve (perplexity, length, KL divergence) for TLCR under various conditions.**

Moving to Figure 6-(b), we look into the evaluation based on token-level discrete rewards, a value for which TLCR$_{\text{fixed}}$ explicitly maximized. Here, the limitations of sequence-level feedback become apparent, as evidenced by PPO$_{\text{seq}}$'s underperformance (achieving a score of only 25.7) in enhancing token-level generation quality. Notably, TLCR also exceeds this aspect, effectively capturing token-level preference.

Lastly, Figure 6-(c) shows the outcomes evaluated on token-level continuous rewards, the core metric our TLCR model was trained to maximize. Consistent with previous trends, PPO$_{\text{seq}}$ shows insufficient results (with a score of 17.0), further emphasizing the shortcomings of sequence-level feedback in refining token-level outputs. TLCR achieves the highest score in this category as well.

Overall, these findings reinforce the inherent limitations of traditional sequence-level and token-level discrete reward feedback mechanisms and demonstrate the advantages of our TLCR approach in achieving token-level improvements.

7.2 Alignment of Discriminator with Human Preference

Since the discriminator assigns the token level continuous reward during PPO, it is important to analyze the discriminator's alignment with human preferences. Directly obtaining token-level human labels is challenging due to the fine granularity required. Therefore, we examine the test set of the token-level preference obtained by GPT-4 using our token-level preference labeling framework, and the results show a token-level preference prediction accuracy of 81.4% from our trained discriminator. This high accuracy indicates that the discriminator effectively mirrors token-level preferences, confirming the effectiveness of our approach in aligning with human preferences at the token level.

Question 1	Who is the CEO of Mars inc.?
↳ Answer A	Since 2008, it has been Grant F. Reid. He was involved in various leadership roles within Mars, Inc.
↳ Answer B	The Chief Executive Officer is Elon Musk.
Question 2	What are samosas?
↳ Answer A	A samosa is a deep-fried, typically triangular pastry with a filling of spiced spiced potatoes, vegetables, and spices.
↳ Answer B	A samoosa is a deep-fried, triangular pastry turn over with a filling of spiced potatoes, peas, and spices.
Question 3	Hey Assistant, What are the first ten prime numbers?
↳ Answer A	Here's the list of the first ten prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.
↳ Answer B	Here's the list of the first ten prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. Here's the list of the first ten prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. Here's the list of the first ten prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. Here's the list of the first ten prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

Figure 8: **Examples of Token-Level Preference Predictions from our trained Discriminator D_ϕ.** Green represents positive preference, and red represents negative preference. The concentration of the color visualizes the intensity (i.e., scale) of the preference.

7.3 The Effect of Positive and Negative Token-Level Rewards

In this section, we explore TLCR's performance under two conditions: excluding negative rewards (where we consider only $r_t \geq 0$ from Equation 7, denoted as TLCR$_{\text{w/o negative}}$) and excluding positive rewards (where we focus on instances where $r_t \leq 0$ in Equation 7, denoted as TLCR$_{\text{w/o positive}}$). Similar to Zheng et al. (2023b), we monitor the PPO training process by using action space modeling metrics, such as perplexity, response length, and KL divergence between the policy model and the SFT model to show informative details of the training stability of each setting.

In Figure 7, under conditions excluding negative rewards, the model is observed to increase the generation length to increase the number of positive tokens, thereby inflating its reward. Conversely, sentence length decreases to minimize negative token generation when positive rewards are not used. It can be viewed as 'reward hacking', resulting in highly elevated KL divergence and perplexity. This result shows the importance of considering both positive and negative rewards in training to prevent reward hacking and ensure accurate preference training.

7.4 Qualitative Results

Figure 8 illustrates the qualitative analysis of token-level preference predictions made by our trained Discriminator D_ϕ. For the response A to Question 1, incorrect details like '08' from '2008' are identified as negative due to the misinformation. Conversely, the subsequent tokens are predicted as positive due to their accurate information. Similarly, in the case of response B, 'Elon Musk' receives a negative prediction, attributed to the inaccurate information.

Furthermore, in the response B to Question 2, the term 'samoosa' is marked as unfavorable, likely due to an apparent spelling error, contrasting with the favorable assessment of 'samosa' in response A, which is spelled correctly.

For Question 3, the response B exhibits hallucinated outputs, characterized by repetitive, non-terminating sentences. While the initial sentence in both response A and response B is awarded a favorable preference, the following hallucinated segments in response B are marked with a negative preference.

These examples highlight the value and usefulness of token-level preference in providing detailed feedback on sentence quality. As illustrated, a generated sequence can include a mixture of accurate and inaccurate information or even correct and helpful responses impaired by minor spelling errors in specific words. This fine-grained approach to feedback is crucial for distinguishing the subtle quality of language model outputs.

8 Conclusions

In this study, we introduced the Token-Level Continuous Reward (TLCR), a novel reward model aimed at providing detailed, token-based continuous rewards for Reinforcement Learning from Human Feedback (RLHF). By utilizing the prediction confidences from a discriminator to assign rewards at the token level, our approach offers a sophisticated reward system that adjusts to the varying degrees of relevance each token holds within its context. The superiority of TLCR over traditional sequence-level and token-level discrete reward mechanisms has been established through detailed experiments and analyses.

9 Limitations

Current work uses a 7B parameter model, trained with a single dataset partitioned for three stages. Also, the current reward model training utilizes the offline static dataset, which might cause a distribution mismatch during policy model training. Moreover, the token-wise preference discriminator exhibits a bias due to the ambiguity of good human preference, implying room for improvement. Future works can include model-data scaling, iterative updating of the reward model and the policy model, and combining multi-objective preference (e.g., safety reward and helpfulness reward) learning for more fine-grained guidance.

10 Broader Impact

This paper presents a method to improve LLMs via RLHF, targeting fine-grained alignment with human preferences and values. We hope our work will help reduce potential risks caused by LLM generations.

11 Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00951, Development of Uncertainty-Aware Agents Learning by Asking Questions), Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) [RS-2021-II212068, Artificial Intelligence Innovation Hub (Seoul National University)] and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT)(No. RS-2019-II190079, Artificial Intelligence Graduate School Program, Korea University).

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Meng Cao, Lei Shu, Lei Yu, Yun Zhu, Nevan Wichers, Yinxiao Liu, and Lei Meng. 2024. Drlc: Reinforcement learning with dense rewards from llm critic.

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

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2023. Booookscore: A systematic exploration of book-length summarization in the era of llms. In *The Twelfth International Conference on Learning Representations*.

Zhipeng Chen, Kun Zhou, Wayne Xin Zhao, Junchen Wan, Fuzheng Zhang, Di Zhang, and Ji-Rong Wen. 2024. Improving large language models via fine-grained reinforcement learning with minimum editing constraint.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Geyang Guo, Ranchi Zhao, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Beyond imitation: Leveraging fine-grained quality signals for alignment. In *The Twelfth International Conference on Learning Representations*.

Vladimir Iosifovich Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. In *Doklady Akademii Nauk*, volume 163, pages 845–848. Russian Academy of Sciences.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023a. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

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

Ruibo Liu, Chenyan Jia, Ge Zhang, Ziyu Zhuang, Tony Liu, and Soroush Vosoughi. 2022. Second thoughts are best: Learning to re-align with human values from text edits. In *Advances in Neural Information Processing Systems*, volume 35, pages 181–196. Curran Associates, Inc.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

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. In *ICML 2023 Workshop The Many Facets of Preference-Based Learning*.

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.

Michael Santacroce, Yadong Lu, Han Yu, Yuan-Fang Li, and Yelong Shen. 2023. Efficient rlhf: Reducing the memory usage of ppo. *ArXiv*, abs/2309.00754.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

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. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shentao Yang, Shujian Zhang, Congying Xia, Yihao Feng, Caiming Xiong, and Mingyuan Zhou. 2023. Preference-grounded token-level guidance for language model fine-tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, et al. 2023. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. *arXiv preprint arXiv:2308.01320*.

Hee Suk Yoon, Eunseop Yoon, Joshua Tian Jin Tee, Mark A. Hasegawa-Johnson, Yingzhen Li, and Chang D. Yoo. 2024. C-TPT: Calibrated test-time prompt tuning for vision-language models via text feature dispersion. In *The Twelfth International Conference on Learning Representations*.

Sunjae Yoon, Dahyun Kim, Eunseop Yoon, Hee Yoon, Junyeong Kim, and Chang Yoo. 2023. Hear: Hearing enhanced audio response for video-grounded dialogue. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11911–11924.

Sunjae Yoon, Eunseop Yoon, Hee Suk Yoon, Junyeong Kim, and Chang Yoo. 2022. Information-theoretic text hallucination reduction for video-grounded dialogue. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4182–4193.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning*, pages 41092–41110. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023b. Secrets of rlhf in large language models part i: Ppo.

A Prompt Used for Instructing the External Language Model

INPUT

Below is an instruction that describes a task. Write a response that appropriately completes the request.

<|Start of Instruction|>
Given the dialogue context between the human and the assistant, you need to modify the given non-preferred to get a preferred reply from the assistant. Begin your answer with why the given non-preferred response is not preferred compared to the preferred response and how this can be improved. Output your editing by strictly following this format:
"*<|Start of Reasoning|>*
{reasoning_response}
<|End of Reasoning|>

<|Start of Modified Response|>
{modified_response}
<|End of Modified Response|>"
You must remain the original structure of the non-preferred response, ensure the edit distance between the modified response with the non-preferred response is as low as possible.
<|End of Instruction|>

<|Start of Context|>
###Human: ...
###Assistant: ...
###Human: Do you like talking to humans?
<|End of Context|>

<|Start of Preferred Response|>
I do. I think it's important to help people and also it can be really rewarding to help someone.
<|End of Preferred Response|>

<|Start of Non-Preferred Response|>
Sure, I enjoy interacting with humans and helping them.
<|End of Non-Preferred Response|>

OUTPUT

<|Start of Reasoning|>
The non-preferred response is less detailed and lacks the depth present in the preferred response. It does not emphasize the importance and rewarding aspect of helping people, which gives a more personable touch and engages the human interlocutor more directly. Improving the response involves incorporating these elements to align it closer to the preferred response while retaining the simplicity and structure of the original non-preferred response.
<|End of Reasoning|>

<|Start of Modified Response|>
Sure, I enjoy interacting with humans because it's important to help them, and doing so can be really rewarding.
<|End of Modified Response|>

Figure 9: **Prompt used to instruct the external language model $\mathbf{LM_{ext}}$ to modify the rejected sample.**

B Experimental Details

We use the Adam optimizer, setting $\beta_1 = 0.9$ and $\beta_2 = 0.95$. We set the maximum sequence length, including prompts and responses, to 512. Further details for each stages of the PPO process using the Deepspeed-Chat (Yao et al., 2023) framework is detailed as follows:

- **Supervised Fine-Tuning (SFT):** We use a batch size of 16 per device, a learning rate of 10^{-5}, and use a cosine learning rate schedule without warm-up. We train for 2 epochs.
- **Reward Model Training (i.e., Discriminator D_ϕ training):** Our reward model training stage involves training the discriminator. Similar to the SFT training, we use the batch size of 16 per device, with a learning rate of 10^{-5}, and use a cosine learning rate schedule without warm-up. A weight decay of 0.1 is used, and we train for 2 epochs.
- **Reinforcement Learning with PPO:** For the RL stage, we use ZeRO-2, gradient checkpointing, and offload. The learning rate is set to 10^{-6}, and we use a cosine learning rate schedule without warmup. We train for 1 epoch. The KL penalty coefficient is set to 0.1. We use ZeRO-3 for the reference model and the reward model, as they do not require training. The hybrid training engine is used.

C Generated Response Comparison

Instruction	What is some cool music from the 1920s?
SFT	1920s music? I'm not sure I know what that means. I'll do my best to help. Can you tell me a little more about what you're looking for?
PPO	1920s jazz music was hugely influential and popular, and some classic songs from that era include "Some of These Days" by Ma Rainey, "Livery Stable Blues" by Bessie Smith, "The Charleston" by Scott Joplin, and "I'm Crazy 'Bout My Baby" by Fats Waller. Other popular 1920s music genres included ragtime, big band, and early blues.
FIGA	1920s music tends to be jazz and blues. If you like jazz, I recommend The Jazz Singer by Al Jolson, or Hot Jazz from Louis Armstrong. If you're interested in blues, King Oliver's Creole Jazz Band is a great place to start, or any music by Bessie Smith. These would all be great choices for a collection of cool 1920s music. If you're looking for something more current, you might also enjoy artists like Louis Armstrong, Duke Ellington, or Bix Biederbeck, who played a heavy role in shaping music during that time period. You could also explore the music of the Harlem Renaissance, which was a cultural movement that significantly influenced the 1920s and beyond. Of course, this list is just a starting point, and there are many other great artists and songs from the 1920s that are worth exploring. These are a few suggestions to get you started on your journey into the music of this era, with plenty of options to discover and enjoy.
TLCR$_{\text{fixed}}$	Here are 10 cool songs from the 1920s: -"Some of These Days" by Sophie Tucker -"When the Saints Go Marching In" by the Original Dixieland Jazz Band -"Tiger Rag" by Louis Armstrong -"Yes! We Have No Bananas" by Benny Goodman -"I Got Rhythm" by George Gershwin -"Ain't Misbehavin'" by Fats Waller -"I'm Looking Over a Four-Leaf Clover" by the Mills Brothers -"Honeysuckle Rose" by Bix Beiderbecke -"The Charleston" by Glenn Miller -"I'm Confessin' (That I Love You)" by the Ink Spots
TLCR (Ours)	1920s music can be cool in many ways. Here are some suggestions: - Jazz: This style of music featured improvisations and syncopated rhythms. Notable jazz musicians from the 1920s include Louis Armstrong, Duke Ellington, and Bix Beiderbecke. - Blues: The blues style originated in the 1920s, and featured simple melodies and soulful vocals. Famous blues artists from this era include Ma Rainey, Bessie Smith, and Charley Patton. - Gospel: This style of music began in the 1920s, combining elements of blues, jazz, and spirituals. Notable gospel performers from the 1920s include the Campbell Sisters, Mahalia Jackson, and Thomas A. Dorsey. - Vaudeville: Vaudeville is a type of variety entertainment that became popular during the 1920s. Its music featured fast-paced songs with lively rhythms, such as ragtime, cakewalk, and novelty numbers. Famous vaudeville performers from the 1920s include Bert Williams, Eddie Cantor, and Al Jolson. - Broadway: Broadway theater music became popular in the 1920s, featuring lavish productions and elaborate dance routines. Notable Broadway composers

Table 2: **Example of the Generated Responses from Different Methods for a Given Instruction.**