# T-REG: Preference Optimization with Token-Level Reward Regularization

**Wenxuan Zhou, Shujian Zhang, Lingxiao Zhao, Tao Meng**
Zoom Communications
wenxuan.zhou@zoom.us

## Abstract

Reinforcement learning from human feedback (RLHF) has been crucial in aligning large language models (LLMs) with human values. Traditionally, RLHF involves generating responses to a query and using a reward model to assign a reward to the entire response. However, this approach faces challenges due to its reliance on a single, sparse reward, which makes it challenging for the model to identify which parts of the sequence contribute most significantly to the final reward. Recent methods have attempted to address this limitation by introducing token-level rewards. However, these methods often rely on either a trained credit assignment model or AI annotators, raising concerns about the quality and reliability of the rewards. In this paper, we propose token-level reward regularization (T-REG), a novel approach that leverages both sequence-level and token-level rewards for preference optimization. Harnessing the self-refinement capabilities of LLMs, our method uses contrastive prompting to enable LLMs to self-generate token-level rewards. These self-generated rewards then act as reward regularization, guiding the model to more effectively distribute sequence-level rewards across tokens. This facilitates better token-level credit assignment and enhances alignment performance. Experiments on the instruction following benchmarks, including Alpaca Eval 2 and Arena-Hard, show that our method consistently outperforms baseline methods by up to 3.8% and 4.4%, respectively.

## 1 Introduction

Recent advancements in large language models (LLMs; Tunstall et al. 2023; Chung et al. 2024; Team et al. 2024) have centered on aligning model outputs with human intentions and preferences. Reinforcement learning from human feedback (RLHF; Christiano et al. 2017; Glaese et al. 2022) has become a dominant approach for achieving this alignment by incorporating human feedback into the training process. The RLHF process typically consists of two phases. In the first phase, responses are generated either with the model being optimized (on-policy RL; Schulman et al. 2017; Guo et al. 2024) or with different models (off-policy RL; Rafailov et al. 2023; Ethayarajh et al. 2024). In the second phase, a reward is obtained based on feedback from either humans (Schulman et al., 2017) or AI annotators (Lee et al., 2023), and this reward is then used to optimize the policy model through algorithms such as PPO (Schulman et al., 2017) and DPO (Rafailov et al., 2023). During this process, the quality of the reward signal is critical to the success of RLHF, where poor or misaligned rewards can lead to issues such as reward hacking (Gao et al., 2023) and model collapse (Wang et al., 2024a; Chowdhury et al., 2024).

Current RLHF methods typically use sequence-level rewards, where reward signals are provided at the end of an entire sequence. Recent studies have shown that incorporating finer-grained reward signals (Wu et al., 2024; Lightman et al., 2023), ultimately at the token level (Zhong et al., 2024), can significantly enhance the performance and convergence of the policy model. However, annotating token-level rewards is challenging for humans, especially when sequences may span thousands of tokens. As a result, current token-level RLHF methods often rely on token-level rewards labeled by AI annotators (Guo et al., 2023; Yoon et al., 2024) or utilize credit assignment models that learn to redistribute sequence-level rewards to individual tokens (Yang et al., 2024b; Zhong et al., 2024). These token-level rewards are then used to train policy models through algorithms such as PPO. However, the quality of these automatically generated rewards remains a concern. AI-generated annotations and credit assignment models are prone to errors, raising questions about the reliability of the rewards and, consequently, the overall quality of the trained policy models.

*Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27876–27889
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

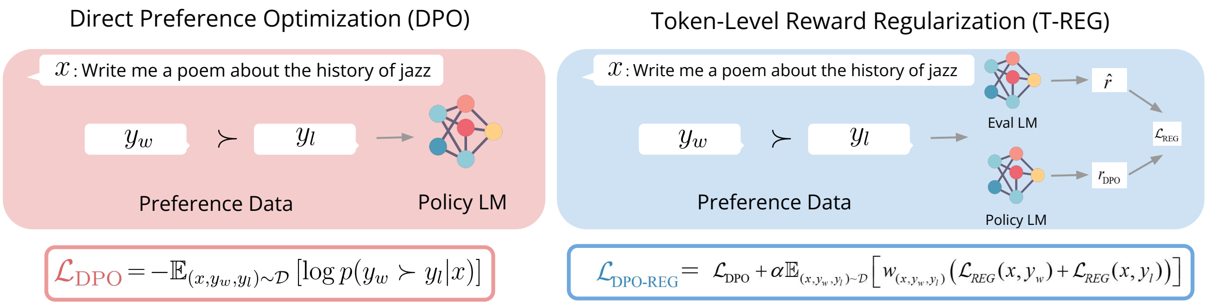
Direct Preference Optimization (DPO)
$x$: Write me a poem about the history of jazz
$y_w$ $\succ$ $y_l$
Preference Data
Policy LM
$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log p(y_w \succ y_l|x)\right]$
Token-Level Reward Regularization (T-REG)
$x$: Write me a poem about the history of jazz
$y_w$ $\succ$ $y_l$
Preference Data
Eval LM
$\hat{r}$
$\mathcal{L}_{\text{REG}}$
$r_{\text{DPO}}$
Policy LM
$\mathcal{L}_{\text{DPO-REG}} = \mathcal{L}_{\text{DPO}} + \alpha\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[w_{(x,y_w,y_l)}\left(\mathcal{L}_{REG}(x,y_w) + \mathcal{L}_{REG}(x,y_l)\right)\right]$

Figure 1: Overview of the Token-Level Reward Regularization (T-REG). Existing DPO directly optimizes the sequence-level rewards to align with user preferences. T-REG prompts LLMs to generate the token-level reward and utilize it as regularization during the preference optimization.

In this paper, we investigate the integration of auto-labeled token-level rewards into preference optimization. We introduce T-REG, a novel token-level preference optimization algorithm that leverages both sequence-level and token-level rewards. Our approach is motivated by the observation that DPO inherently learns token-level rewards. Instead of directly optimizing the policy model with auto-labeled token-level rewards, T-REG retains sequence-level preference optimization while using these token-level rewards as weak supervision. This weak supervision regularizes the token-level reward signals implicitly learned during preference optimization. This design enables T-REG to optimize the overall sequence-level reward while also achieving effective token-level credit assignment. Inspired by the self-refinement capabilities of LLMs (Madaan et al., 2024), we derive token-level rewards through contrastive prompting, where revision-based prompts rewrite the output in opposing directions. By comparing the token probabilities generated from these prompts, we compute the token-level rewards. This approach assigns token-level rewards effectively without requiring additional training or token-level annotations.

We evaluate T-REG on two instruction following benchmarks, including Alpaca Eval 2 (Li et al., 2023) and Arena-Hard (Li et al., 2024). T-REG demonstrates consistent improvements across both Llama-3-Instruct-8B and Gemma-2-9B-it models. On Alpaca Eval 2, T-REG increases the length-controlled win rate by up to 24.8% over the SFT checkpoint, surpassing DPO by up to 3.8%. On the more challenging Arena-Hard benchmark, T-REG improves the win rate by up to 20.0%, outperforming DPO by as much as 4.4%. Moreover, we find that T-REG can be integrated into different preference optimizations (e.g., SimPO) and shows consistent improvements. Additionally, we analyze the token-level rewards learned by T-REG, demonstrating that it effectively achieves token-level credit assignment.

Our contributions are summarized as follows:

- We propose T-REG, a token-level preference optimization method that utilizes both sequence-level and token-level rewards to enhance alignment with human preferences.
- T-REG utilizes token-level rewards derived through contrastive prompting to guide the token-level rewards learned during preference optimization, enabling effective token-level credit assignment without the need for external token-level reward annotations.
- T-REG achieves consistent improvements on instruction-following benchmarks including Alpaca Eval 2 and Arena-Hard, outperforming DPO by up to 3.8% and 4.4%, respectively, and showing compatibility with other preference optimization algorithms.

## 2 Related Work

**Reinforcement learning from human feedback.** Reinforcement Learning from Human Feedback (RLHF) has become a cornerstone in aligning Large Language Models (LLMs) with human intentions, allowing for fine-tuning of models to produce more useful and aligned responses (Ziegler et al., 2019). The early implementations of RLHF typically used the proximal policy optimization algorithm (PPO; Schulman et al. 2017), which operates by optimizing policies through episodic rewards, calculated using a reward model trained on human preferences (Bai et al., 2022; Ouyang et al., 2022). However, applying deep RL techniques such as PPO to LLMs has proven challenging due to instability and inefficiency in training (Engstrom et al., 2019). To address these challenges, alternative methods have emerged that employ supervised fine-tuning (SFT) to simulate the RL process, using high-quality responses to create more stable training signals (Lu et al., 2022; Zhao et al., 2023; Yang et al., 2024b). In contrast to classical RLHF methods, a new line of research has proposed direct alignment algorithms (Zhao et al., 2023; Rafailov et al., 2023; Meng et al., 2024; Azar et al., 2024; Zhou et al., 2024a; Wang et al., 2024b), which optimize a supervised target directly based on preference data, without the need for a separate reward model. Despite their advantages, these methods rely solely on sequence-level rewards. In this work, we extend this paradigm by prompting LLMs to generate token-level rewards using contrastive prompting, integrating these rewards as regularization during preference optimization.

**Algorithm 1:** Token-Level Reward Regularization (T-REG)

```
Input: Dataset ($\mathcal{D}$) with prompts and responses, policy model $\pi_\theta$, total number of iterations $T$, learning rate $\alpha_t$,
for $t = 0$ to $T$ do
    Sample a mini-batch of tuples $(x, y_w, y_l)$ from $\mathcal{D}$,
    Generate token-level rewards via Eq. (3),
    Define the regularization at token-level rewards via Eq. (1),
    Compute $\mathcal{L}_{\text{DPO-REG}}$ via Eq. (2),
    Update policy parameters $\theta$ using gradient descent: $\theta \leftarrow \theta - \alpha_t \nabla\theta(x, y_w, y_l, \theta)$.
end for
```

**Token-level RLHF.** While sequence-level rewards provide effective supervision for RLHF, the extended length of LLM generations means that different parts of a sequence contribute unequally to the final rewards, making sequence-level rewards less data-efficient and effective (Chan et al., 2024; Zhong et al., 2024). To address these limitations, recent efforts have shifted from sequence-level to token-level rewards, with a focus on how to effectively derive these finer-grained signals. Existing methods have adopted various approaches to token-level reward derivation. For instance, Token-Level Direct Preference Optimization (TDPO; Zeng et al. 2024) uses Markov decision processes (MDPs) to derive token-level rewards. Other methods, such as token-level continuous reward (TLCR; Yoon et al. 2024), FIGA (Guo et al., 2023), and RLMEC (Chen et al., 2024), rely on off-the-shelf LLMs to auto-label token-level rewards. Additionally, since DPO has been shown to implicitly learn token-level rewards (Rafailov et al., 2023), some approaches (Zhong et al., 2024; Yang et al., 2024a) leverage DPO for this purpose. Once token-level rewards are derived, existing methods typically optimize the policy using algorithms such as PPO (Zhong et al., 2024; Yoon et al., 2024) or token selection strategies (Yang et al., 2024a). However, these approaches often depend heavily on auto-labeled token-level rewards, which introduces potential challenges due to the inherent noise of auto-labeled token-level rewards. In our approach, we use token-level rewards derived from LLMs as guidance in preference optimization to enhance token-level credit assignment. Therefore, we mitigate the risks of solely relying on auto-labeled token-level rewards, achieving a more robust and effective optimization process.

## 3 Method

In this section, we present in detail our proposed method, T-REG (Algorithm 1). We first provide the theoretical background of RLHF in Section 3.1. Next, we introduce the token-level credit assignment problem and explain how to guide preference optimization using a token-level regularization in Section 3.2. Finally, we present our contrastive prompting method for generating token-level rewards in Section 3.3.

### 3.1 Preliminaries

**RLHF.** Reinforcement learning from human feedback (RLHF; Schulman et al. 2017) is a widely used technique for aligning LLMs with human preferences. Using human or AI annotations, RLHF enables LLMs to generate outputs that align better with human expectations. Consider a preference dataset $\mathcal{D} = \left(x^{(i)}, y_w^{(i)}, y_l^{(i)}\right)_{i=1}^{N}$, where each sample consists of a prompt $x^{(i)}$ and two outputs, $y_w^{(i)}$ (preferred) and $y_l^{(i)}$ (dispreferred), as judged by annotators, the core of RLHF lies in a reward model, $r^*(x, y)$, which assigns a score to each candidate output $y$ based on how well it aligns with the prompt $x$. A commonly adopted framework for modeling preference distributions is the Bradley-Terry (BT; Bradley and Terry 1952) model, which defines the probability that $y_w$ is preferred over $y_l$ as:

$$p(y_w \succ y_l|x) = \frac{\exp(r^*(x,y_w))}{\exp(r^*(x,y_w))+\exp(r^*(x,y_l))}.$$

The RLHF process typically consists of two phases: learning the reward model from preference data, followed by using reinforcement learning to optimize a policy model based on this reward. In the first phase, the reward model is trained using maximum likelihood estimation, producing an estimated reward function $\hat{r}(x, y)$. The reward model can be structured to return feedback either at the end of a sequence (Liu et al., 2023; Dong et al., 2023; Xiong et al., 2024), where the evaluation is based on the entire output, or at each step of the sequence (Uesato et al., 2022; Lightman et al., 2023), where feedback is provided based on intermediate reasoning steps. Once the reward model is trained, it is used to finetune the policy model by optimizing the following objective:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)} \left[\hat{r}(x, y) - \beta \log \frac{\pi_\theta(\cdot|x)}{\pi_{\text{ref}}(\cdot|x)}\right],$$

where $\beta$ is a KL penalty coefficient to regularize the deviation between the policy model $\pi_\theta$ and the reference model $\pi_{\text{ref}}$.

**DPO implicitly learns token-level rewards.** In RLHF, rewards are typically assigned at the end of sequences, which can extend to thousands of tokens. However, not all tokens contribute equally to the final reward (Chan et al., 2024; Rafailov et al., 2024; Yang et al., 2024a). To address this, it is crucial for a preference optimization algorithm to effectively distribute sequence-level rewards across individual tokens. Previous work (Zhong et al., 2024; Rafailov et al., 2024) has shown that Direct Preference Optimization (DPO; Rafailov et al. 2023) implicitly learns token-level rewards under sequence-level reward supervision. Specifically, with the token-level reward defined as $r^*_{\text{token}}(y_t|x, y_{<t}) = \beta \log \frac{\pi^*(y_t|x,y_{<t})}{\pi_{\text{ref}}(y_t|x,y_{<t})}$, the sequence-level reward is computed as[1]:

$$r_{\text{DPO}}(x, y) = V^*(x) + \sum_{t=1}^{T} \beta \log \frac{\pi^*(y_t|x, y_{<t})}{\pi_{\text{ref}}(y_t|x, y_{<t})},$$

where $T$ is the number of tokens in the sequence, $\pi^*$ is the optimal policy, $\pi_{\text{ref}}$ is the reference policy, and $V^*$ is the value function of $\pi^*$. For a pair of outputs $(y_w, y_l)$, the probability that $y_w$ is preferred over $y_l$, modeled using the BT model, is given by:

$$\begin{aligned} p(y_w \succ y_l|x) =& \sigma \left( \sum_{t=1}^{T_w} \beta \log \frac{\pi^*(y_{w_t}|x, y_{w_{<t}})}{\pi_{\text{ref}}(y_{w_t}|x, y_{w_{<t}})} \right. \\ & \left. - \sum_{t=1}^{T_l} \beta \log \frac{\pi^*(y_{l_t}|x, y_{l_{<t}})}{\pi_{\text{ref}}(y_{l_t}|x, y_{l_{<t}})} \right), \end{aligned}$$

where $V^*(x)$ cancels out because $y_w$ and $y_l$ correspond to the same prompt $x$. By applying maximum likelihood estimation to the preference dataset, the policy model $\pi$ can be optimized by the following loss function:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log p(y_w \succ y_l|x)\right],$$

which resembles the original DPO loss (Rafailov et al., 2023). Therefore, performing RLHF with DPO is redistributing the sequence-level reward into the token level as $\beta \log \frac{\pi(y_t|x,y_{<t})}{\pi_{\text{ref}}(y_t|x,y_{<t})}$, where $\pi$ is the policy trained by DPO. This redistribution ensures token-level credit assignment while optimizing the sequence-level objective.

### 3.2 Regularized Token-level Preference Optimization

**Token-level reward as regularization.** In preference optimization, our objective is to ensure consistency between the model's pairwise rankings and human preference while also achieving effective token-level credit assignment to enhance the model's generalization capabilities. Since DPO models are trained with sequence-level rewards as supervision, they can capture pairwise rankings at the sequence level (Lambert et al., 2024). However, since token-level rewards are implicitly derived through the redistribution of sequence-level rewards in DPO, they often lack direct guidance. Recent studies have demonstrated that LLMs can serve as dense token-level reward functions without fine-tuning, by employing techniques such as contrastive decoding (Li et al., 2022) or contrastive prompting (Kim et al., 2024; Zhao et al., 2024b). These methods infer token-level rewards from the difference in token probabilities between a strong model and a weak model, expressed as $\log \frac{\pi_{\text{strong}}(y_t|x,y_{<t})}{\pi_{\text{weak}}(y_t|x,y_{<t})}$ (see details in Section 3.3). However, these methods do not guarantee that the accumulated sequence-level rewards align with the ground-truth sequence-level rewards provided by the preference data. In this paper, we propose integrating token-level and sequence-level rewards

[1] For a detailed derivation, refer to Rafailov et al. (2024).

**Prompting Template for Revising an Output**

**Instruction:** Below is a conversation between an user and an AI Assistant.
[User Question]
{instruction}
[The start of Assistant's Answer]
{answer}
[The end of Assistant's Answer]
Please rewrite the Assistant's Answer to make it {direction}. Specifically, the rewritten {direction} answer should closely resemble the original answer but is {direction} in terms of one or multiple of the following aspects: helpfulness, correctness, coherence, verbosity.
IMPORTANT: Please strictly follow the following format: First, choose one or multiple aspects to generate a {direction} answer, such as rewrite the original answer to be {detailed_description}, etc.
[The start of a rewritten {direction} answer]
<provide a {direction} answer here>
[The end of a rewritten {direction} answer]

Figure 2: Prompt template for revising the output to either a better or worse one. To make the output *better*, we set use "helpful, correct, coherent, concise" as the description. To make the output *worse*, we use "unhelpful, incorrect, incoherent, verbose" as the description.

to leverage the strengths of both. Our approach incorporates token-level rewards as regularization in preference optimization to improve token-level credit assignment, thereby improving generalization of the policy model.

Specifically, we introduce a regularization term to ensure that the token-level rewards learned by DPO align with dense token-level rewards derived from LLMs. Let $r_{\text{token}}$ denote the token-level reward learned by policy $\pi$, and $\hat{r}_{\text{token}}$ denote the given dense token-level reward, we can compute their similarity at $y_t$ by:

$$\text{sim}(y_t \mid x, y_{<t}) = r_{\text{token}}(y_t \mid x, y_{<t})\, \hat{r}_{\text{token}}(y_t \mid x, y_{<t}).$$

Our goal is to maximize the alignment between the token-level rewards across the entire output. The regularization term, therefore, is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{reg}} &= -\sum_{t=1}^{T} \text{sim}(y_t \mid x, y_{<t}) \\ &= -\textstyle\sum_{t=1}^{T} \beta \hat{r}_{\text{token}}(y_t \mid x, y_{<t}) \, \log \frac{\pi(y_t|x,y_{<t})}{\pi_{\text{ref}}(y_t|x,y_{<t})}.\end{aligned}$$

Here, the term $\pi_{\text{ref}}(y_t|x, y_{<t})$ is a constant reference probability, which does not affect the gradient and can thus be omitted. This leads to the following modified regularization term:

$$\mathcal{L}_{\text{reg}} = -\textstyle\sum_{t=1}^{T} \beta \hat{r}_{\text{token}}(y_t \mid x, y_{<t}) \log \pi(y_t|x, y_{<t}). \quad (1)$$

This term acts as a weighted language modeling loss, increasing the likelihood of tokens with positive token-level rewards while decreasing the likelihood of tokens with negative token-level rewards, thereby improving token-level credit assignment.

**Regularized token-level preference optimization.** During preference optimization, we then optimize both the preference optimization loss (e.g., $\mathcal{L}_{\text{DPO}}$) and the regularization loss ($\mathcal{L}_{\text{reg}}$). The preference optimization loss aims to increase the margin between the probabilities of preferred and dispreferred sequences, while the regularization loss maximizes the probabilities of high-reward tokens. Since these two objectives optimize in different directions, balancing them is critical for effective optimization. To achieve this balance, we use sequence-level gradient norms (Chen et al., 2018). Recall that the gradient of the DPO loss is:

$$\begin{aligned}\nabla \mathcal{L}_{\text{DPO}} = -\beta\sigma\left(r_{\text{DPO}}(x, y_l) - r_{\text{DPO}}(x, y_w)\right) \\ \left(\nabla \log \pi\left(y_w|x\right) - \nabla \log \pi\left(y_l|x\right)\right).\end{aligned}$$

where for each sequence, the gradient norm with respect to each token is proportional to $\sigma\left(r_{\text{DPO}}(x, y_l) - r_{\text{DPO}}(x, y_w)\right)$. To ensure that the regularization loss does not dominate the preference optimization loss for each sequence, we introduce a sequence weight to modulate the regularization loss. The final regularized token-level preference optimization objective is:

$$\begin{aligned}\mathcal{L}_{\text{DPO-REG}} = \mathcal{L}_{\text{DPO}} + \alpha \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[w_{(x,y_w,y_l)}\right. \\ \left.\left(\mathcal{L}_{\text{REG}}(x, y_w) + \mathcal{L}_{\text{REG}}(x, y_l)\right)\right],\end{aligned} \quad (2)$$

where $w_{(x,y_w,y_l)} = \sigma\left(r_{\text{DPO}}(x, y_l) - r_{\text{DPO}}(x, y_w)\right)$ is the sequence weight and is *detached* from back propagation, $\alpha$ is a hyperparameter that controls the strength of regularization. As shown in our ablation study (see Section 4.2), sequence weighting achieves the best performance.

### 3.3 Self-generated Token-level Rewards

In this section, we focus on deriving the dense token-level rewards by LLMs. Existing work (Zhou et al., 2024b; Zhao et al., 2024a) often approximates this reward by contrasting a strong model against a weak model, where:

$$\hat{r}(y_t|x, y_{<t}) = \log \frac{\pi_{\text{strong}}(y_t|x, y_{<t})}{\pi_{\text{weak}}(y_t|x, y_{<t})}.$$

The strong model can be a model from the same family but at a larger scale (Li et al., 2022), a model that is more aligned with human expectations (Zhou et al., 2024b; Zhong et al., 2024; Huang et al., 2024), or a model with better prompts (Kim et al., 2024; Zhao et al., 2024b). In this paper, we focus on using *contrastive prompting* to derive token-level reward, as this approach only requires a single model and does not require additional fine-tuning. Specifically, we utilize two contrastive, revision-based prompts, $x_{\text{better}}$ and $x_{\text{worse}}$ designed to refine the current output $y$ in positive or negative directions to evaluate token quality. Building on prior work in self-refinement (Madaan et al., 2024), we leverage the demonstrated ability of LLMs to adjust outputs in diverse directions. The revision prompt, adapted from Wang et al. (2024e) and illustrated in Fig. 2, refines an output based on four aspects: helpfulness, correctness, coherence, and verbosity. Given a token $y_t$ and a causal language model $\pi_{\text{eval}}$ for generating the rewards, we calculate the token-level reward by:

$$\hat{r}(x, y_{<t}, y_t) = \sigma \left( \log \frac{\pi_{\text{eval}}(y_t|x_{\text{better}}, y_{<t})}{\pi_{\text{eval}}(y_t|x_{\text{worse}}, y_{<t})} \right) - 0.5, \quad (3)$$

where $\sigma$ clips the reward value, and $-0.5$ clips recenters the original reward into range of $[-0.5, 0.5]$. This normalization helps mitigate extreme token-level reward values, thereby stabilizing the preference optimization process. Thanks to the autoregressive nature of causal language models, the rewards for all tokens in an output can be derived with only two forward passes. Due to tokenization issues, $\pi_{\text{eval}}$ should ideally share the same vocabulary as the reference model $\pi_{\text{ref}}$[2]. In this work, we specifically focus on the self-generated token-level rewards, where we use the reference model $\pi_{\text{ref}}$ to generate the token-level rewards.

## 4 Experiments

In this section, we outline our experimental settings in Section 4.1, present the main results and ablations in Section 4.2, and provide qualitative case studies in Section 4.3. Additional analysis of T-REG can be found in Appendix A.

### 4.1 Experimental Settings

**Model configurations.** Our methods are implemented using the official repo of Zephyr[3]. Preference optimization is performed based on two large language models: Llama-3-8B-Instruct (Dubey et al., 2024) and Gemma-2-9B-it (Team et al., 2024). We use the hyperparameters used by Meng et al. (2024), who conducted an extensive hyperparameter search. For our newly introduced hyperparameter, $\alpha$, we search in the range $\{0.1, 0.25, 0.5\}$.

**Training data.** We perform RLHF in an on-policy setting, where outputs are sampled from the policy being optimized. However, generating outputs during training is computationally expensive. To address this, we adopt an approximate method similar to that used by Meng et al. (2024), where outputs are sampled using the reference policy before preference optimization. Specifically, the reference policy generates five outputs for each prompt, which are then evaluated using an external reward model. The best and worst outputs are selected to form a preference pair. For our experiments, we use the preference data generated by Meng et al. (2024), derived from prompts in Ultrafeedback (Cui et al., 2023) and scored using the ArmoRM reward model (Wang et al., 2024d,c).

**Evaluation.** We evaluate the models on two benchmarks: Alpaca Eval 2 (Li et al., 2023) and Arena-Hard (Li et al., 2024). Alpaca Eval 2 is an automated benchmark designed to assess the alignment of LLMs with human preferences across 805 representative instructions. For each instruction, the evaluated model's response is compared head-to-head with the response generated by `gpt-4-turbo` using an automatic evaluator (with `gpt-4-turbo` as the default evaluator). The win rate reflects the probability that the evaluator prefers the responses of the evaluated model over those of `gpt-4-turbo`. Additionally, Alpaca Eval 2 introduces a length-controlled win rate (Dubois et al., 2024) to mitigate length bias in `gpt-4-turbo`. We use the generation configurations recommended by Zheng et al. (2024) to generate the outputs during evaluation.

Arena-Hard is an automated benchmark consisting of 500 user queries that emphasize challenging topics, such as coding and expert-level knowledge.

[2] Although there are methods to resolve mismatches in vocabulary (Kasai et al., 2022), we consider models with the same vocabulary for simplicity.

[3] https://github.com/huggingface/alignment-handbook

| Method | Llama-3-Instruct (8B) | | | Gemma-2-Instruct (9B) | | |
|---|---|---|---|---|---|---|
| | Alpaca Eval 2.0 | | Arena-Hard | Alpaca Eval 2.0 | | Arena-Hard |
| | Len-control. Win Rate | Win Rate vs GPT-4 | Win Rate vs GPT-4 | Len-control. Win Rate | Win Rate vs GPT-4 | Win Rate vs GPT-4 |
| SFT | 26.0 | 25.3 | 22.3 | 51.1 | 38.1 | 40.8 |
| RTO | 49.2 | 47.2 | 37.6 | 67.6 | 68.7 | 63.1 |
| SePO | 48.5 | 45.5 | 33.1 | - | - | -[4] |
| TDPO$_1$ | 45.4 | 37.4 | 29.1 | 66.5 | 56.6 | 46.7 |
| TDPO$_2$ | 43.2 | 40.2 | 34.0 | 64.3 | 59.3 | 55.7 |
| DPO | 47.0 | 46.0 | 35.9 | 68.9 | 66.9 | 58.6 |
| SimPO | 52.5 | 47.1 | 33.1 | 73.5 | **70.7** | 63.0 |
| DPO-REG | $\underline{50.8}$ | $\underline{\mathbf{51.1}}$ | $\underline{\mathbf{40.3}}$ | $\underline{70.3}$ | 66.4 | 60.2 |
| SimPO-REG | $\underline{\mathbf{53.8}}$ | $\underline{48.8}$ | 34.4 | $\underline{\mathbf{74.5}}$ | 70.5 | $\underline{\mathbf{64.2}}$ |

Table 1: Evaluation results (%) on Alpaca Eval 2 and Arena-Hard benchmarks. Scores that are underlined denote statistically significant gains ($p < 0.05$).

The model's response is compared head-to-head with the response generated by `gpt-4-0314`. Notably, Arena-Hard exhibits the highest correlation with Chatbot Arena and offers the best separability among widely used open-ended LLM benchmarks. In our experiments, we use the official generation configurations provided by Arena-Hard.

**Compared methods.** We consider two preference optimization algorithms, DPO and SimPO (Meng et al., 2024)[5], as the base algorithms for applying T-REG. Additionally, we evaluate the following token-level preference optimization algorithms for comparison: (1) **RTO.** Reinforced token optimization (Zhong et al., 2024) first trains a DPO model to derive the dense token-level rewards. It then uses PPO (Schulman et al., 2017) to directly optimize the policy model based on these token-level rewards. (2) **SePO.** Selective preference optimization (Yang et al., 2024a) also derives token-level rewards using DPO. It then performs preference optimization on a subset of tokens by selecting those with the highest reward values in the preferred output and the lowest reward values in the dispreferred output. (3) **TDPO.** Token-level direct preference optimization (Zeng et al., 2024) optimizes preferences at the token level and includes forward KL divergence constraints for each token. This approach has two variants, TDPO$_1$ and TDPO$_2$. This method does not utilize explicit token-level rewards as supervision. As these methods do not report their numbers on our base models, we reproduce them and search for the best set of hyperparameters within their recommended range.

## 4.2 Main Results and Ablation

**T-REG consistently outperforms the baselines and the compared methods.** The results on Alpaca Eval 2 and Arena-Hard are presented in Table 1. We observe that T-REG, when applied to both DPO and SimPO, consistently outperforms these methods on both benchmarks. Specifically, on Alpaca Eval 2, T-REG increases the length-controlled win rate by up to 24.8% over the SFT checkpoint, surpassing DPO by up to 3.8%. On the more challenging Arena-Hard benchmark, T-REG improves the win rate by up to 20.0%, outperforming DPO by as much as 4.4%. Similar improvements are observed with SimPO, indicating that although T-REG is primarily derived from DPO, it generalizes well to other preference optimizations. Among other token-level preference optimization methods, RTO, which conducts PPO on token-level rewards derived from DPO, is the best performing one and achieves results comparable to or better than DPO, especially on the Arena-Hard benchmark. However, RTO's gains are smaller or even negative on Alpaca Eval 2, which focuses on general questions, while T-REG consistently yields positive improvements across both benchmarks. These results highlight the effectiveness of T-REG in enhancing preference optimization.

**Selective regularization on high-reward tokens yields better results.** As shown in Eq. 1, T-REG can be interpreted as performing weighted SFT on high-reward tokens. Previous methods (Dubey et al., 2024; Xu et al., 2024) apply an SFT loss over the entire preferred output $y_w$, which has been shown to help prevent degeneration:

$$\mathcal{L}_{\text{DPO-SFT}} = \mathcal{L}_{\text{DPO}} + \alpha \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \pi(y_w|x)\right].$$

[4] Despite extensive hyperparameter tuning, we find that Gemma-2-Instruct, fine-tuned by SePO, suffers severely from degeneration and fails to produce reasonable results.

[5] While T-REG is derived based on DPO, it can be adapted to other preference optimization algorithms by adjusting the sequence-level weighting term according to their gradients.

| Method | Alpaca Eval 2.0 | | Arena-Hard |
|---|---|---|---|
| | Len-control. Win Rate | Win Rate vs GPT-4 | Win Rate vs GPT-4 |
| DPO | 47.0 | 46.0 | 35.9 |
| DPO-REG | 50.8 | 51.1 | 40.3 |
| DPO-SFT on $y_w$ | 46.0 | 43.4 | 32.7 |
| Static weigh. | 48.0 | 46.3 | 35.1 |
| DPO reward | 49.8 | 51.0 | 36.9 |

Table 2: Results of different variants of T-REG based on Llama-3-Instruct (8B).

However, as shown in Table 2, incorporating this loss results in significant performance degradation on both tasks, even with a small $\alpha$ value. This is likely because optimizing over the entire $y_w$ includes tokens of lower quality. In contrast, T-REG — focuses exclusively on high-reward tokens — achieves consistent improvements over DPO.

**Sequence weighting enhances performance.** To evaluate the impact of sequence-level weighting on performance, we trained DPO-REG using static weighting by removing the weighting term from the loss. However, as shown in Table 2, this approach did not result in consistent improvements over DPO, showing the effectiveness of sequence-level weighting.

**Self-generated reward outperforms DPO reward.** In line with prior approaches like RTO and SePO, we also experimented with using the token-level rewards derived from DPO for regularization. As shown in Table 2, this approach achieved similar performance to self-generated rewards on Alpaca Eval 2 but performed much worse on Arena-Hard, underperforming self-generated rewards by 3.4%. These results indicate that self-generated token-level rewards are comparable to or better than those derived from DPO.

### 4.3 Case Study

In this section, we examine the quality of token-level rewards learned by T-REG. Since no existing evaluation datasets specifically assess token-level rewards, we perform a qualitative analysis instead. Figure 3 presents three example tokens from Chatbot Arena (Chiang et al., 2024). We calculate the token-level rewards with $\log \frac{\pi_\theta(y_t|x,y_{<t})}{\pi_{\text{ref}}(y_t|x,y_{<t})}$, where $\pi_\theta$ and $\pi_{\text{ref}}$ are the policy and reference models, respectively. The positive and negative rewards are visualized in red and blue. Our analysis demonstrates that integrating T-REG into DPO enhances the precision of token-level reward assignments.

- **Prompt 1:** The prompt specifies that only the initial letter of the word "Test" should be capitalized, but the response is entirely in uppercase. Here, T-REG assigns a negative reward to capture the mismatch, whereas DPO incorrectly assigns a positive reward.
- **Prompt 2:** The correct response is *one dog*, yet DPO erroneously assigns a positive reward to the incorrect answer. In contrast, T-REG accurately reflects the error by assigning a negative reward.
- **Prompt 3:** The correct answer is 4, and the response matches. While DPO fails to assign a positive reward to this response, T-REG correctly assigns a positive reward.

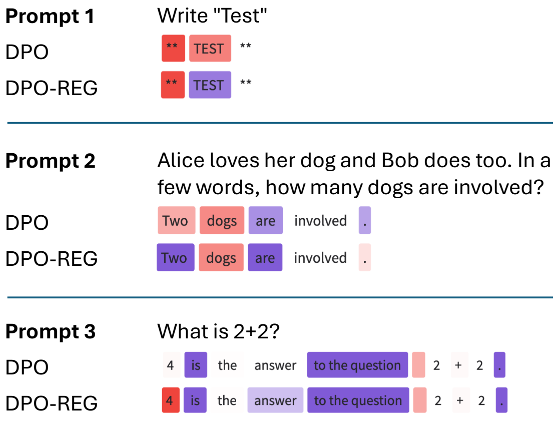
Prompt 1
Write "Test"
DPO
** TEST **
DPO-REG
** TEST **
Prompt 2
Alice loves her dog and Bob does too. In a few words, how many dogs are involved?
DPO
Two dogs are involved .
DPO-REG
Two dogs are involved .
Prompt 3
What is 2+2?
DPO
4 is the answer to the question 2 + 2 .
DPO-REG
4 is the answer to the question 2 + 2 .

Figure 3: Case study on the token-level rewards learned by DPO and DPO-REG, where red means positive reward and blue means negative reward. We use Llama-3-Instruct (8B) as the base model.

These examples underscore the effectiveness of T-REG in achieving precise token-level reward assignments.

## 5 Conclusion

In this study, we addressed the challenge of token-level credit assignment in preference optimization by introducing T-REG, a novel method that leverages self-generated token-level rewards derived through contrastive prompting as regularization. This approach allows for more effective and fine-grained credit assignment, seamlessly integrating with sequence-level preference optimization to enhance alignment with human preferences. Experiments on instruction-following benchmarks including Alpaca Eval 2 and Arena-Hard, as well as our qualitative case study, demonstrate that T-REG not only enhances alignment performance but also achieves more precise token-level credit assignment.

## Limitations

**Lack of quantitative results on token-level credit assignment.** In this paper, we only provide qualitative results on the token-level rewards learned by T-REG. Therefore, there lacks a systematic and more rigorous study on the accuracy of token-level rewards. While we believe this analysis will provide valuable insights, to the best of our knowledge, no benchmarks currently exist for evaluating token-level rewards. Future work should focus on constructing dedicated evaluation datasets to facilitate quantitative assessment of token-level reward accuracy.

**Rewards in other levels.** Our approach focuses on utilizing rewards at the token and sequence levels. However, intermediate levels, such as step-level and span-level rewards, also provide useful information for alignment tasks and have been widely applied, especially in math and coding problems. The current method does not account for these intermediate reward levels. Future research could explore methods that incorporate multiple levels of rewards, potentially enhancing the flexibility and effectiveness of preference optimization.

## References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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

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*.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR.

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. *arXiv preprint arXiv:2401.06081*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024. Provably robust dpo: Aligning language models with noisy feedback. *arXiv preprint arXiv:2403.00409*.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2019. Implementation matters in deep rl: A case study on ppo and trpo. In *International conference on learning representations*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. 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*.

Geyang Guo, Ranchi Zhao, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Beyond imitation: Leveraging fine-grained quality signals for alignment. *arXiv preprint arXiv:2311.04072*.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. 2024. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*.

James Y Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024. Offset unlearning for large language models. *arXiv preprint arXiv:2404.11045*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Hao Peng, Ximing Lu, Dragomir Radev, Yejin Choi, and Noah A Smith. 2022. Twist decoding: Diverse generators guide each other. *arXiv preprint arXiv:2205.09273*.

Taehyeon Kim, Joonkee Kim, Gihun Lee, and Se-Young Yun. 2024. Instructive decoding: Instruction-tuned large language models are self-refiner from noisy instructions. In *The Twelfth International Conference on Learning Representations*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

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

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

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

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

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. 2024. 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. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

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

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. 2024a. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Fei Wang, Wenxuan Zhou, James Y Huang, Nan Xu, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024b. mdpo: Conditional preference optimization for multimodal large language models. *arXiv preprint arXiv:2406.11839*.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. 2024c. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. In *ACL*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024d. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024e. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2024. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2024. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *Preprint*, arXiv:2312.11456.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. In *Forty-first International Conference on Machine Learning*.

Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Erxue Min, and Sophia Ananiadou. 2024a. Selective preference optimization via token-level reward function estimation. *arXiv preprint arXiv:2408.13518*.

Shentao Yang, Shujian Zhang, Congying Xia, Yihao Feng, Caiming Xiong, and Mingyuan Zhou. 2024b. Preference-grounded token-level guidance for language model fine-tuning. *Advances in Neural Information Processing Systems*, 36.

Eunseop Yoon, Hee Suk Yoon, SooHwan Eom, Gunsoo Han, Daniel Wontae Nam, Daejin Jo, Kyoung-Woon On, Mark A Hasegawa-Johnson, Sungwoong Kim, and Chang D Yoo. 2024. Tlcr: Token-level continuous reward for fine-grained reinforcement learning from human feedback. *arXiv preprint arXiv:2407.16574*.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*.

Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang, and William Yang Wang. 2024a. Weak-to-strong jailbreaking on large language models. *arXiv preprint arXiv:2401.17256*.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Zhengyue Zhao, Xiaoyun Zhang, Kaidi Xu, Xing Hu, Rui Zhang, Zidong Du, Qi Guo, and Yunji Chen. 2024b. Adversarial contrastive decoding: Boosting safety alignment of large language models via opposite prompt optimization. *arXiv preprint arXiv:2406.16743*.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. 2024. Weak-to-strong extrapolation expedites alignment. *arXiv preprint arXiv:2404.16792*.

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*.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. 2024a. Wpo: Enhancing rlhf with weighted preference optimization. *arXiv preprint arXiv:2406.11827*.

Zhanhui Zhou, Zhixuan Liu, Jie Liu, Zhichen Dong, Chao Yang, and Yu Qiao. 2024b. Weak-to-strong search: Align large language models via searching over small language models. *arXiv preprint arXiv:2405.19262*.

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

| Method | IFEval | BHH | MATH-Hard | GPQA | MUSR | MMLU-Pro | Average |
|---|---|---|---|---|---|---|---|
| **Llama-3-Instruct (8B)** | | | | | | | |
| SFT | 74.08 | 28.24 | 8.69 | 1.23 | 1.60 | 29.60 | 23.91 |
| DPO | 65.13 | 26.53 | 4.75 | 5.47 | 8.77 | 28.72 | 23.23 |
| DPO-REG | 68.69 | 29.16 | 7.70 | 6.97 | 5.17 | 31.44 | 24.86 |
| SimPO | 71.74 | 28.01 | 4.16 | 7.36 | 8.05 | 29.63 | 24.82 |
| SimPO-REG | 67.78 | 27.30 | 5.13 | 5.17 | 8.06 | 29.44 | 23.85 |
| **Gemma-2-Instruct (9B)** | | | | | | | |
| SFT | 74.36 | 42.14 | 0.23 | 14.77 | 9.74 | 31.95 | 28.86 |
| DPO | 71.43 | 41.11 | 0.00 | 15.88 | 7.96 | 29.09 | 27.58 |
| DPO-REG | 70.67 | 42.10 | 0.00 | 14.53 | 6.46 | 31.29 | 27.51 |
| SimPO | 73.29 | 41.14 | 0.00 | 14.56 | 10.00 | 33.03 | 28.67 |
| SimPO-REG | 70.56 | 41.06 | 0.00 | 14.36 | 10.10 | 32.69 | 28.13 |

Table 3: Results on the OpenLLM leaderboard v2.

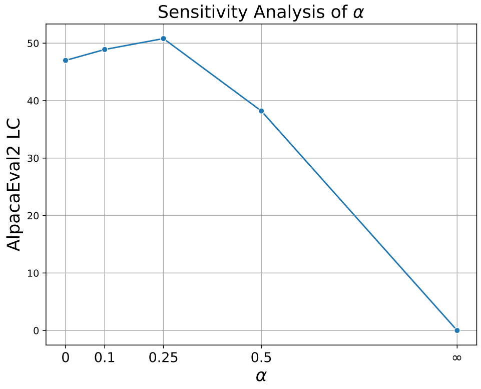

Figure 4: Performance on Alpaca Eval 2 across various $\alpha$ values using Llama-3-Instruct.

## A Additional Analysis

**Results on downstream tasks.** We evaluate the models on the OpenLLM leaderboard v2 (Fourrier et al., 2024) to assess their capabilities on general downstream tasks. We use the lm-evaluation-harness[6], the official code base for the OpenLLM leaderboard v2. Results are shown in Table 3. Overall, we observe that RLHF models exhibit performance on par with, or slightly below, that of the SFT model on downstream tasks. Similarly, a comparison between T-REG and the baseline models reveals no significant performance differences. These findings align with previous research (Askell et al., 2021), which suggests that improved alignment does not necessarily correlate with enhanced general task performance.

**Sensitivity analysis of $\alpha$.** In our experiments, we focus solely on tuning the value of $\alpha$, while adopting all other hyperparameters directly from prior work. To evaluate its impact, we present a sensitivity analysis of $\alpha$ in Figure 4. The results indicate consistent performance improvements for smaller $\alpha$ values, such as $0.1$ and $0.25$, but a decline as $\alpha$ increases, ultimately resulting in degeneration as $\alpha$ approaches $\infty$. These findings underscore the importance of balancing sequence-level and token-level rewards, with sequence-level rewards playing a more significant role, as evidenced by the small optimal $\alpha$ values.

## B Additional Details

**Scientific artifacts.** This work utilizes various scientific resources, including LLMs, preference datasets, and evaluation tools or benchmarks. We provide references to all the artifacts used, where details about their licenses, supported languages, data coverage, parameter sizes, and any safety considerations can be

[6] https://github.com/EleutherAI/lm-evaluation-harness

accessed. Given the broad scope of data types and domains covered by modern LLMs and preference datasets, we do not detail these specifics here but encourage readers to refer to the original sources for more comprehensive information. All resources in this study were used strictly for non-distributive and non-commercial research purposes, adhering to the terms of their respective licenses.

**Budget.** We conduct all experiments using 8 × H100 GPUs. The experiments take approximately 4 hours for Llama-3-Instruct and 7 hours for Gemma-2-it.

**Use of AI assistants.** We employed ChatGPT exclusively to refine the language and improve the readability of the paper. The tool was used solely for stylistic enhancements and clarity improvements, without influencing the scientific content or analysis.