# CREAM: CONSISTENCY REGULARIZED SELF-REWARDING LANGUAGE MODELS

**Zhaoyang Wang**[1] **Weilei He**[2] **Zhiyuan Liang**[3] **Xuchao Zhang**[4]
**Chetan Bansal**[4] **Ying Wei**[2] **Weitong Zhang**[1] **Huaxiu Yao**[1]
[1]University of North Carolina at Chapel Hill [2]Nanyang Technological University
[3]National University of Singapore [4]Microsoft Research
{zhaoyang,huaxiu}@cs.unc.edu weitongz@unc.edu

## ABSTRACT

Recent self-rewarding large language models (LLM) have successfully applied LLM-as-a-Judge to iteratively improve the alignment performance without the need of human annotations for preference data. These methods commonly utilize the same LLM to act as both the policy model (which generates responses) and the reward model (which scores and ranks those responses). The ranked responses are then used as preference pairs to train the LLM via direct alignment technologies (e.g. DPO). However, it is noteworthy that throughout this process, there is no guarantee on the accuraty of the rewarding and ranking, which is critical for ensuring accurate rewards and high-quality preference data. Empirical results from relatively small LLMs (e.g., 7B parameters) also indicate that improvements from self-rewarding may diminish after several iterations in certain situations, which we hypothesize is due to accumulated bias in the reward system. This bias can lead to unreliable preference data for training the LLM. To address this issue, we first formulate and analyze the generalized iterative preference fine-tuning framework for self-rewarding language model. We then introduce the regularization to this generalized framework to mitigate the overconfident preference labeling in the self-rewarding process. Based on this theoretical insight, we propose a **C**onsistency **R**egularized s**E**lf-rewarding l**A**nguage **M**odel (CREAM) that leverages the consistency of rewards across different iterations to regularize the self-rewarding training, helping the model to learn from more reliable preference data. With this explicit regularization, our empirical results demonstrate the superiority of CREAM in improving both reward consistency and alignment performance.

## 1 INTRODUCTION

Large language models (LLMs) have shown impressive capabilities across various tasks, including natural language understanding and generation (Radford et al., 2019). At the same time, LLMs also face alignment challenges such as generating hallucinations and harmful outputs (Ji et al., 2023). To address these issues, a series of research works have explored preference learning methods such as Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017) and direct alignment techniques such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) to align the LLMs with human values and preferences. These alignment methods often require a large amount of preference pairs which are indispensable in both RLHF and direct alignment training. However, collecting human-annotated preference pairs is time-consuming and labor-intensive, which seriously limits the scalability and efficiency of these alignment methods.

Recent advancements in self-rewarding language models (SRLMs) (Yuan et al., 2024) have gained increasing attention in the field of LLM alignment, which can efficiently synthesize preference data for iterative preference training. In this method, the single LLM is required to act as two roles, the policy model and the reward model. Given unlabeled prompt data, the LLM first acts as the policy model generating several response candidates. Then, the same LLM acts as the reward model, scoring and ranking these responses. These ranked responses are used as preference pairs to train the LLM with DPO, significantly reducing the reliance on human-annotated data. The above steps can be iteratively repeated to further enhance the performance. However, SRLMs still face challenges

in generating reliable and accurate rewards for annotating the preference pairs, which is critical for ensuring the quality of preference data and the alignment performance of LLMs.

To address these challenges, we first formulate a generalized iterative preference fine-tuning framework to analyze the self-rewarding training, where this framework can also be adapted to other iterative preference tuning methods. Through this theoretical framework, we find that the rewarding bias issue in SRLMs comes from the overconfident preference labeling, which enforces the model to distinguish between responses with similar quality. For example, both two responses in Figure 1 have high quality judgments from the human. The SRLM enforces the reward model to make a preference judgment, resulting in noisy and unreliable preference labeling. This can lead to negative impacts on preference tuning the model. Additionally, the iterative training manner can also accumulate the rewarding bias, further diminishing the benefits of self-improvement. From the insights of theoretical analysis, we propose **C**onsistency **R**egularized s**E**lf-rewarding l**A**nguage **M**odel (CREAM) to mitigate the rewarding bias issue in SRLMs, particularly for broadly accessible 7B-size LLMs. The core idea behind CREAM is that we should not force the model to be overly confident in distinguishing between responses with similar quality. *But how to tell the preference labeling is reliable or not?* Out of the self-rewarding scenario, we may employ a pool of external reward models to assist in ranking preferences. When two responses are of similar quality, these external models often produce inconsistent rankings. This inconsistency serves as a signal to indicate the level of confidence in the preference labeling. In self-rewarding scenarios, however, integrating such external reward models is not feasible. Fortunately, due to the iterative nature of self-rewarding training, we can use the reward model from the previous iteration to rank preferences and then compare these rankings with those produced by the current model. This comparison provides an estimate of such consistency rate. With this consistency rate, we can regularize the preference training to prevent the model from learning unreliable preference data, thereby mitigating the rewarding bias issue in SRLMs.

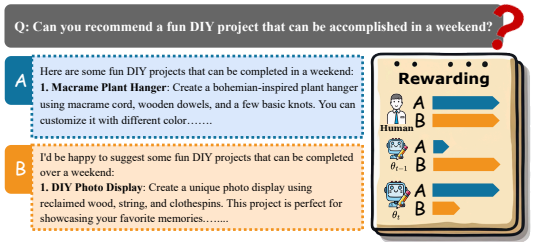


Figure 1: An example of both two responses are of high quality, which is hard for human to distinguish the preference. While the same model from different iterations have inconsistent rewarding.

In summary, we first formulates a generalized iterative preference fine-tuning framework to analyze the rewarding bias issue in SRLMs. From the insights of theoretical analysis, we propose CREAM as the primary contribution of this paper. CREAM leverages the consistency of rewards across different iterations for regularized preference training, which can effectively mitigate the rewarding bias issue in SRLMs. Empirical results on a series of natural language benchmarks validate the effectiveness of CREAM in mitigating the rewarding bias issue and enhancing the alignment performance of LLMs.

**Notations.** Vectors are denoted by lowercase boldface letters, such as $\mathbf{x}$, and matrices by uppercase boldface letters, such as $\mathbf{A}$. For any positive integer $k$, the set $1, 2, \ldots, k$ is denoted by $[k]$. Sets are denoted by calligraphic uppercase letters, such as $\mathcal{D}$, with the cardinality of the set represented as $|\mathcal{D}|$. Without ambiguity, we denote $\pi_{\boldsymbol{\theta}}$ as the language model parameterized by $\boldsymbol{\theta}$, $\mathbf{x}$ as the input prompt, and $\mathbf{y}$ as the output response from the language model. All other notations are defined prior to their first usage. We denote $\mathbb{1}[\cdot]$ as the indicator function.

## 2 Methodology

In this section, we first formulate the generalized iterative preference fine-tuning framework for self-rewarding, RL with AI feedback, and other iterative preference tuning methods. Next, we introduce the motivation behind the proposed consistency regularized self-rewarding method. Finally, we present the practical implementation algorithm of CREAM in details.

### 2.1 Generalized Iterative Preference Fine-Tuning Framework

We assume that we can access to the dataset with response $\mathcal{D}_{\text{S}}$ and the prompt dataset without response $\mathcal{D}_{\text{U}}$. The objective is to iteratively minimize the following loss with respect to the neural

network parameter $\boldsymbol{\theta}$ and a label function $z$ as

$$\mathcal{L}(\boldsymbol{\theta}, z) = \mathcal{L}_{\text{SFT}}(\boldsymbol{\theta}; \mathcal{D}_{\text{S}}) + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{U}};\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}[\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x}, z)]. \tag{2.1}$$

where the first term $\mathcal{L}_{\text{SFT}}(\boldsymbol{\theta}; \mathcal{D}_{\text{S}})$ aligns the model $\pi_{\boldsymbol{\theta}}$ to the SFT data. We note here that any potential SFT methods (Ouyang et al., 2022b; Yuan et al., 2023; Dong et al., 2023; Chen et al., 2024), or the methods without SFT data ($\mathcal{L}_{\text{SFT}} = 0$) can be adapted in this framework. The second term $\mathbb{E}[\mathcal{L}_{\text{DPO}}]$ corresponds to learning from the preference data pair $\{\mathbf{y}, \mathbf{y}'\}$ generated by the current model $\boldsymbol{\theta}_t$. The labeling function $z(\mathbf{y}, \mathbf{y}', \mathbf{x}) \in \{0, 1\}$ provides the preference judgment between $\mathbf{y}$ and $\mathbf{y}'$ for the DPO loss, where $z(\mathbf{y}, \mathbf{y}', \mathbf{x}) = 1$ means $\mathbf{y} \succ \mathbf{y}'$ and $z(\mathbf{y}, \mathbf{y}', \mathbf{x}) = 0$ means $\mathbf{y} \prec \mathbf{y}'$. The DPO loss $\mathcal{L}_{\text{DPO}}$ is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x}, z) = &-z(\mathbf{y}, \mathbf{y}', \mathbf{x}) \log \sigma \left( \log \left( \frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})} \right) - \log \left( \frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})} \right) \right) \\ &- (1 - z(\mathbf{y}, \mathbf{y}', \mathbf{x})) \log \sigma \left( \log \left( \frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})} \right) - \log \left( \frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})} \right) \right), \end{aligned} \tag{2.2}$$

where $\pi_{\text{ref}}$ is the reference model for KL divergence regularization, and $\sigma(\cdot)$ is the sigmoid function. The proposed loss $\mathcal{L}(\boldsymbol{\theta}, z)$ in Eq. (2.1) represents all iterative preference fine-tuning algorithms. For the reinforcement learning (RL) with human feedback (Ouyang et al., 2022b), $z$ is the human preference comparing $\mathbf{y}$ and $\mathbf{y}'$. For the RL with AI feedback, $z$ is the oracle reward model like GPT-4 (Achiam et al., 2023). For the self-rewarding language model (Chen et al., 2024), $z$ is given by comparing the reward score generated from the language model itself, often with LLM-as-a-Judge prompting. However, as aforementioned, we note that such prompt rewarding method may only be feasible for larger and advanced LLMs such as Llama-70B (Touvron et al., 2023). For smaller models such as Llama-7B that do not have complex instruction following and reasoning abilities, we instead propose to leverage the intrinsic reward model (Rafailov et al., 2023)

$$r_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) \propto [\log \pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) - \log \pi_{\text{ref}}(\mathbf{y}|\mathbf{x})]$$

to reward and rank the responses for annotating preference pairs. Therefore, the choice of preference labeling function $z$ is closely connected with the language model parameter $\boldsymbol{\theta}$. Then, we introduce the following two-step optimization algorithm to solve Eq. (2.1).

**Step 1.** (Preference-labeling step) Keep $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ fixed, select function $z$ to minimize $\mathcal{L}_{\text{DPO}}$. In particular, letting $\boldsymbol{\theta} = \boldsymbol{\theta}_t$ in Eq. (2.2), solution for $z(\mathbf{y}, \mathbf{y}', \mathbf{x}) = \arg\min_z \mathcal{L}_{\text{DPO}}(\boldsymbol{\theta}_t; \mathbf{y}, \mathbf{y}', \mathbf{x}, z)$ is

$$z_{t+1}(\mathbf{y}, \mathbf{y}', \mathbf{x}) = \mathbb{1}\left[\log \pi_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) - \log \pi_{\text{ref}}(\mathbf{y}|\mathbf{x}) \geq \log \pi_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x}) - \log \pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})\right]. \tag{2.3}$$

**Step 2.** (Learning step) Keep $z$ as of Eq. (2.3), minimize loss function $\mathcal{L}(\boldsymbol{\theta}, z_{t+1})$ with respect to $\boldsymbol{\theta}$ and get $\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, z_{t+1})$.

Different from existing methods, the proposed two-step optimization method directly uses the intrinsic reward model to generate the preference data. This approach is particularly feasible for smaller LLMs, which lack the capacity to effectively use LLM-as-a-Judge prompts (Zheng et al., 2023) for rewarding and ranking. We note that the proposed two-step method is similar to the Expectation-Maximization algorithm and self-training paradigm (Zou et al., 2019). This similarity is supported by the following theorem, which suggests the convergence of the proposed two-step algorithm.

**Theorem 2.1.** Suppose the optimization $\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, z_{t+1})$ is solvable and the SFT loss $\mathcal{L}_{\text{SFT}}(\boldsymbol{\theta}; \mathcal{D}_{\text{S}}) \geq 0$ for all $\boldsymbol{\theta}$ and $\mathcal{D}_{\text{S}}$, the proposed two-step optimization method converges.

### 2.2 Consistency Regularized Self-Rewarding

The generalized framework presented in Eq. (2.1) assumes the human feedback or GPT-4 are all reliable so that the preference labeling function $z$ is trustworthy. However, for SRLMs, the accuracy of preference labeling is not always guaranteed. Therefore, treating all selected preference labels as "ground truth" by encoding them as hard labels can lead to overconfident mistakes, potentially propagating biases and inaccuracies from the LLMs. Taking Figure 1 as an example, both the two responses $\mathbf{y}$ and $\mathbf{y}'$ are judged by humans to be of high quality. *Forcing the model to be overly confident in distinguishing between these two responses $\{\mathbf{y}, \mathbf{y}'\}$ with similar quality can negatively impact the performance of SRLMs during training.*

This rewarding bias issue motivates us to mitigate such ambiguity by introducing a consistency-regularized self-rewarding language model, CREAM. Specifically, for a pair of responses with very similar quality, their oracle reward scores should ideally be very close to each other. Particularly, when multiple reward models are available, it is likely that some models will rank one response as superior, while others may rank the opposite response as better, resulting in high ranking inconsistency (i.e., low ranking consistency) among these models. Based on this, CREAM aims to prevent the model from learning from preference pairs with low consistency. Instead, it focuses solely on preference pairs with high consistency across different reward models, thereby mitigating the rewarding bias issue and stabilize the learning process to some extent. From the theoretical perceptive, we can introduce a regularization term to Eq. (2.1) as

$$\mathcal{L}(\boldsymbol{\theta}, z) = \mathcal{L}_{\text{SFT}}(\boldsymbol{\theta}; \mathcal{D}_{\text{S}}) + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{U}};\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}[\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x}, z) + \lambda\mathcal{L}_{\text{Reg}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x})], \quad (2.4)$$

where the regularization term $\mathcal{L}_{\text{Reg}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x})$ prevents the model $\pi_{\boldsymbol{\theta}}$ from overconfidence in distinguishing the preference of $\{\mathbf{y}, \mathbf{y}'\}$ with similar quality, which is quantified in the following lemma.

**Lemma 2.2.** Let the random variable $z = z(\mathbf{y}, \mathbf{y}', \mathbf{x})$ be defined as $z(\mathbf{y}, \mathbf{y}', \mathbf{x}) = \mathbb{1}[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}]$. The Bradley-Terry model (Bradley & Terry, 1952) for the probability of $z$ under parameter $\boldsymbol{\theta}$ is given by

$$P_{\boldsymbol{\theta}}(z) = P_{\boldsymbol{\theta}}(\mathbb{1}[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}]) = \sigma\left(\log(\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})/\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})) - \log(\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})/\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x}))\right),$$

Letting the regularization $\mathcal{L}_{\text{Reg}}$ be defined by

$$\begin{aligned}\mathcal{L}_{\text{Reg}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x}) = &-\log\sigma\left(\log(\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})/\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})) - \log(\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})/\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x}))\right) \\ &-\log\sigma\left((\log\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})/\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})) - (\log\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})/\pi_{\text{ref}}(\mathbf{y}|\mathbf{x}))\right). \quad (2.5)\end{aligned}$$

Then the expected regularized loss under the model $\boldsymbol{\theta}_t$ is given by:

$$\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\mathcal{L}_{\text{reg}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x}) = 2\,\mathbb{KL}(u(\cdot) \parallel P_{\boldsymbol{\theta}}(\cdot)), \quad (2.6)$$

where $u(z)$ is the uniform binary distribution, i.e., $u(z = 0) = u(z = 1) = 0.5$.

As Lemma 2.2 suggests, the $\mathcal{L}_{\text{Reg}}$ will regularize the preference between $\{\mathbf{y}, \mathbf{y}'\}$ that has similar quality to a uniform distribution. Then the following theorem suggests that using $\mathcal{L}_{\text{DPO}} + \lambda\mathcal{L}_{\text{Reg}}$ corresponds to the soft-labeled DPO which we implemented in CREAM.

**Theorem 2.3.** For all $\mathbf{y}, \mathbf{y}', \mathbf{x}, z$, minimizing

$$\mathcal{L}(\boldsymbol{\theta}, z) = \mathcal{L}_{\text{SFT}}(\boldsymbol{\theta}; \mathcal{D}_{\text{SFT}}) + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{U}};\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x}, z) + \lambda\mathcal{L}_{\text{Reg}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x})\right]$$

is equivariant with minimizing

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, z) = &\frac{1}{1+2\lambda}\mathcal{L}_{\text{SFT}}(\boldsymbol{\theta}; \mathcal{D}_{\text{S}}) \\ &+ \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{U};\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}[\mathcal{C}_\lambda\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x}, z) + (1-\mathcal{C}_\lambda)\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{y}', \mathbf{x}, 1-z)], \quad (2.7)\end{aligned}$$

where the $1 - z$ reverses the preference order of $z(\mathbf{y}, \mathbf{y}', \mathbf{x})$ and $\mathcal{C}_\lambda = (1 + \lambda)/(1 + 2\lambda)$.

Theorem 2.3 suggests that instead of calculating the regularization term $\mathcal{L}_{\text{Reg}}$, we can use the soft-labeled DPO to train Eq. (2.7). In particular, when $\lambda = 0$, $\mathcal{C}_\lambda = 0$ and Eq. (2.7) degenerates to Eq. (2.1). This represents the case where the preference label $z$ is trustworthy from human or some oracle reward models (e.g., GPT-4). In other words, $\lambda$ represents the *confidence* of the label function $z$. Specially, since in our two-step optimization paradigm, the label function $z$ is directly derived from the previous model $\pi_{\boldsymbol{\theta}_t}$, we can measure the performance of $\pi_{\boldsymbol{\theta}_t}$ using the consistency between model $\boldsymbol{\theta}_t$ and the baseline model (e.g., external reward model) $\boldsymbol{\theta}'_t$, defined by

$$\lambda(\mathbf{x}) = 2\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\,\mathbb{1}[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}, \boldsymbol{\theta}'_t], \quad (2.8)$$

and when $\lambda \to 0$, $\mathcal{C}_\lambda \approx 1 - \lambda$ representing the consistency of model $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}'_t$. $\mathbb{1}[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}, \boldsymbol{\theta}_t]$ means the response $\mathbf{y}$ is better than $\mathbf{y}'$ given the prompt $\mathbf{x}$ and language model parameter $\boldsymbol{\theta}_t$, i.e.,

$$\mathbb{1}[\log(\pi_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})/\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})) - \log(\pi_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})/\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x}))],$$

and similar definition applies to $\mathbb{1}[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}, \boldsymbol{\theta}'_t]$.

**Algorithm 1** Consistency-Regularized Self-Rewarding Language Model

**Input:** seed SFT dataset $\mathcal{D}_{\text{S}}$; unlabeled prompt dataset $\mathcal{D}_{\text{U}}$; initial model parameter $\boldsymbol{\theta}_0$;
**Input:** number of iterations $T$; learning rate $\eta$
1: /* SFT training */
2: Obtain $\boldsymbol{\theta}_1$ by taking the gradient steps over loss $L_1(\boldsymbol{\theta}) = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_{\text{S}}} \log \pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ from $\boldsymbol{\theta}_0$
3: /* Iterative Preference Training training */
4: **for** $t = 1$ to $T$ **do**
5: Sample $\{\mathbf{y}_{ij}\}_{i=1}^{N} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)$ for all $\mathbf{x}_j \in \mathcal{D}_{\text{U}}$ // Response Sampling
6: Compute reward $r_{ij} = \log \pi_{\boldsymbol{\theta}_t}(\mathbf{y}_{ij}|\mathbf{x}_i) - \log \pi_{\boldsymbol{\theta}_0}(\mathbf{y}_{ij}|\mathbf{x}_i)$ for all $i \in [N], j \in [|\mathcal{D}_{\text{U}}|]$
7: Obtain rank $J_{ij}$ for all $y_{ij}$ using reward $r_{ij}$ // Rank on model $\boldsymbol{\theta}_t$
8: Compute reward $r'_{ij} = \log \pi_{\boldsymbol{\theta}_{t-1}}(\mathbf{y}_{ij}|\mathbf{x}_i) - \log \pi_{\boldsymbol{\theta}_0}(\mathbf{y}_{ij}|\mathbf{x}_i)$ for all $i \in [N], j \in [|\mathcal{D}_{\text{U}}|]$
9: Obtain rank $K_{ij}$ for all $y_{ij}$ using reward $r'_{ij}$ // Rank on model $\boldsymbol{\theta}_{t-1}$
10: Compute $\tau_j = \tau(\{J_{ij}\}_i, \{K_{ij}\}_i)$ according to Eq. (2.10) for all $j \in [|\mathcal{D}_{\text{U}}|]$
11: Compute consistency rate $\mathcal{C} = |\mathcal{D}_{\text{U}}|^{-1} \sum_j (\tau_j + 1)/2$ // Adaptive consistency regularization
12: Compose preference dataset $\mathcal{D}_{\text{DPO}}$ using pairs $\{\mathbf{x}_j, \mathbf{y}_j^+, \mathbf{y}_j^-\}_j$ according to Eq. (2.11)
13: Compose preference dataset $\mathcal{D}_{\text{RDPO}}$ using pairs $\{\mathbf{x}_j, \mathbf{y}_j^-, \mathbf{y}_j^+\}_j$ according to Eq. (2.12)
14: Update $\boldsymbol{\theta}_{t+1}$ by minimizing loss $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{C}\mathcal{L}_{\text{DPO}}(\pi_{\boldsymbol{\theta}_t}, \mathcal{D}_{\text{DPO}}) + (1-\mathcal{C})\mathcal{L}_{\text{DPO}}(\pi_{\boldsymbol{\theta}_t}, \mathcal{D}_{\text{RDPO}})$
15: **end for**
**Output:** aligned policy model $\pi_{\boldsymbol{\theta}_T}$

### 2.3 Proposed Algorithm

Equipped with the above two-stage optimization and the consistency-regularized self-rewarding, we are ready to present the implementation of CREAM in Algorithm 1. The whole framework of CREAM is also illustrated in Figure 2. The algorithm starts from the SFT training to obtain the first model parameter $\boldsymbol{\theta}_1$ in Line 2. A similar approach is applied in Yuan et al. (2024) for avoid calculating the $\mathcal{L}_{\text{SFT}}$ in the future optimization steps. Then for each $\mathbf{x}_j$ in the unlabeled prompt set $\mathcal{D}_{\text{U}}$, $N$ response candidates $\{\mathbf{y}_i\}_{i=1}^{N}$ are sampled in Line 5. Then reward scores of these $N$ candidates can be calculated according to Rafailov et al. (2023) by

$$r_{ij} = \beta[\log \pi_{\boldsymbol{\theta}_t}(\mathbf{y}_{ij}|\mathbf{x}_j) - \log \pi_{\boldsymbol{\theta}_0}(\mathbf{y}_{ij}|\mathbf{x}_j)] + \beta \log Z(\mathbf{x}_j), \tag{2.9}$$

where we use the initial model parameter $\boldsymbol{\theta}_0$ as the reference policy $\pi_{\text{ref}}$. Since $\beta \geq 0$ and $\log Z(\mathbf{x}_j)$ is a constant across different response $\mathbf{y}_i$ for the same input prompt $\mathbf{x}_j$, we can drop these factors and calculate rewards in Line 6. Specially, when $t = 1$, the rank $K_{ij}$ is calculated based on the reference policy $\boldsymbol{\theta}_0$ itself. Thus we instead use the likelihood $r_{ij} = \log \pi_{\boldsymbol{\theta}_0}(\mathbf{y}_{ij}|\mathbf{x}_j)$ as the reward for this edge case. The rank for these $N$ candidates are therefore obtained in Line 7, where $J_{ij}$ means response $\mathbf{y}_{ij}$ is in the $J_{ij}$-th best in the preference list of $\mathbf{x}_j$.

**Consistency-Regularized Self-Rewarding.** As discussed in Eq. (2.8), a baseline model is required to measure the consistency. In the self-rewarding scenario, it is infeasible to add an external reward model as the baseline model. Fortunately, we can employ the model before last update $\boldsymbol{\theta}_{t-1}$ as the baseline model $\boldsymbol{\theta}'_t$ (i.e., last iteration's model) for evaluating the consistency of the model $\boldsymbol{\theta}$, thanks to chances provided by iterative training manner. Such a procedure helps mitigate the training error introduced in $t-1$-th step before obtaining $\boldsymbol{\theta}_t$. Considering a pair of tied preference pair $\mathbf{y}, \mathbf{y}'$ both performing well, as demonstrated in Figure 1. $P[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}, \boldsymbol{\theta}_t]$ will be oscillating around $0.5$ when $t$ grows due to the random noise. Otherwise $P[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}, \boldsymbol{\theta}_t]$ might consistently converge to $0$ or $1$. Due to this oscillation, the consistency between $\boldsymbol{\theta}_{t-1}$ and $\boldsymbol{\theta}_t$ on this specific preference pair would be low, and the algorithm will learn less from this noninformative preference pair thus stabilize this oscillation.

Specifically, we calculate the rank of these $N$ candidates using $\boldsymbol{\theta}_{t-1}$ in Line 9 and then use the Kendall's Tau coefficient (Kendall, 1938) denoted by

$$\tau_j = \frac{2}{N(N-1)} \sum_{1\leq i<i'\leq N} \left[\mathbb{1}\left[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) > 0\right] - \mathbb{1}\left[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) < 0\right]\right]. \tag{2.10}$$

Kendall's Tau coefficient is a widely used coefficient (McLeod, 2005; Abdi, 2007) to measure the consistency of two ranking sequences. Basically, when two sequences perfectly aligns, $\tau_j = 1$ and

when two sequence never aligns, $\tau_j = -1$. The following lemma draws the further connection between the Kendall's Tau and the regularization parameter $\lambda$ proposed in Section 2.2.

**Lemma 2.4.** Suppose the $N$ response candidate $\{\mathbf{y}_{ij}\}_i$ is i.i.d. given the prompt $\mathbf{x}_j$, then

$$\mathbb{E}[\tau_j] = 1 - 4\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)} \mathbb{1}[\mathbf{y} \succ \mathbf{y}'|\mathbf{x}_j, \boldsymbol{\theta}_t] \mathbb{1}[\mathbf{y} \prec \mathbf{y}'|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}] = 1 - 2\lambda,$$

where the expectation is taken over the randomness of sampling the $N$ candidate set.

Given Lemma 2.4, we can recover $\mathcal{C}_\lambda \approx 1 - \lambda = (1 + \tau_j)/2$ and we use average all $\tau_j$ for all $\mathbf{x}_j \in \mathcal{D}_{\text{U}}$ in Line 11. Finally, in Line 12, we compose the preference dataset by selecting the best response $\mathbf{y}_j^+ = \mathbf{y}_{i^+j}$ and the worst response $\mathbf{y}_j^- = \mathbf{y}_{i^-j}$ which is similar with (Yuan et al., 2024).

$$\mathcal{D}_{\text{DPO}} = \{(\mathbf{x}_j, \mathbf{y}_{i^+j}, \mathbf{y}_{i^-j})|\mathbf{x}_j \in \mathcal{D}_{\text{U}}, i^+ = \arg\min_i J_{ij}, i^- = \arg\min_i J_{ij}\} \tag{2.11}$$

Following Theorem 2.3, we also prepare the reverse DPO dataset by switching the best response and the worst response by

$$\mathcal{D}_{\text{RDPO}} = \{(\mathbf{x}_j, \mathbf{y}_{i^-j}, \mathbf{y}_{i^+j})|\mathbf{x}_j \in \mathcal{D}_{\text{U}}, i^+ = \arg\min_i J_{ij}, i^- = \arg\min_i J_{ij}\} \tag{2.12}$$

and update $\boldsymbol{\theta}_{t+1}$ by minimizing the empirical loss of Eq. (2.7) in Line 14. The detailed proof of theorems and lemmas are provided in the Appendix B.

# 3 Experiment

## 3.1 Experimental Setup

**Data.** In our experiments, we use Open Assistant dataset (Köpf et al., 2024) and only reserve about 3.4K human-annotated examples as the seed SFT data $\mathcal{D}_{\text{S}}$. To construct the unlabeled prompt dataset $\mathcal{D}_{\text{U}}$, we mix prompts of $\mathcal{D}_{\text{S}}$ with the train split of each downstream task (only reserve the prompts) including (1) ARC-Easy/Challenge (Clark et al., 2018), (2) OpenBookQA (Mihaylov et al., 2018), (3) SIQA (Sap et al., 2019), and (4) GSM8K (Cobbe et al., 2021). Finally, this process results in a total of 21K prompts in $\mathcal{D}_{\text{U}}$, which we distribute equally across iterative self-rewarding trainings.

**Models.** Due to limited computational resources, we mainly conduct experiments with two LLMs with about 7B parameters, including Llama-3 (Dubey et al., 2024) and Llama-2 (Touvron et al., 2023).

**Baseline Methods.** To validate our findings, we mainly compare our method with SRLM (Yuan et al., 2024) which uses the same LLM to serve as both the policy and reward model to generated preference data for iterative training. Additionally, we introduce a variant of RL with AI feedback (Guo et al., 2024a), referred to as "Oracle". In this variant, the reward model in SRLM is replaced with an external reward model to demonstrate the upper bound performance of SRLM. Specifically, we use InternLM2 (Cai et al., 2024), a specialized trained reward model, to provide the reward scores for the generated responses. We further enhance Oracle's rewarding by leveraging the labels from downstream tasks to improve the rewarding accuracy.

**Implementation Details.** In our experiments, we fine-tune the initial model (M0) on the seed SFT data for 3 epochs with a learning rate of $1e-6$, resulting in model M1. Following SRLM approach, we then iteratively fine-tune the model using the preference learning objective two additional iterations, producing models M2 and M3. In the preference training of each iteration, we set $\beta = 0.1$ of DPO loss, and fine-tune the model for 1 epoch with a learning rate of $1e-6$.

## 3.2 Main Results

The main results are shown in Table 1 which also report the performance of GPT-4o for reference. From these results, we observe the following: (1) The Standard SRLM fails to achieve satisfactory performance, particularly with Llama2 which has relatively weaker foundation performance even after SFT fine-tuning (M0 $\rightarrow$ M1), which indicates its limitations for 7B-level LLMs. (2) Compared to SRLM, CREAM achieves a significant improvement across almost all downstream tasks, showing the advantage of introducing the proposed regularization method. (3) SRLM equipped with an

Table 1: Main results of each method on test sets of downstream tasks. The exact match accuracies are reported. The "↑" and "↓" indicate the performance improvement and degradation compared to the method's last iteration (e.g., M1 → M2 and M2 → M3), respectively. The best performance between SRLMs and CREAM is highlighted in bold.

| Model | Method | Arc-Easy | Arc-Challenge | OpenBookQA | SIQA | GSM8K |
|---|---|---|---|---|---|---|
| GPT-4o | CoT | 94.57 | 94.71 | 96.60 | 79.63 | 92.27 |
| Llama-3 | M0 | 86.29 | 80.37 | 86.00 | 68.58 | 78.01 |
| | M1 | 86.78 | 80.14 | 86.40 | 69.50 | 78.39 |
| | Oracle M2 | 89.60 ↑ | 82.17 ↑ | 90.00 ↑ | 72.88 ↑ | 80.82 ↑ |
| | Oracle M3 | 89.31 ↓ | 81.31 ↓ | 90.20 ↑ | 73.75 ↑ | 76.04 ↓ |
| | SRLM M2 | 87.79 ↑ | 80.38 ↑ | 87.80 ↑ | 70.95 ↑ | 78.01 ↓ |
| | SRLM M3 | 87.17 ↓ | 81.23 ↑ | 87.30 ↓ | 70.37 ↓ | 77.48 ↓ |
| | CREAM M2 | 88.89 ↑ | 80.89 ↑ | 88.00 ↑ | 69.79 ↑ | 81.04 ↑ |
| | CREAM M3 | **89.52** ↑ | **83.36** ↑ | **90.20** ↑ | **72.06** ↑ | **81.73** ↑ |
| Llama-2 | M0 | 61.07 | 48.98 | 62.20 | 50.36 | 23.65 |
| | M1 | 60.44 | 48.46 | 63.20 | 50.77 | 23.88 |
| | Oracle M2 | 70.20 ↑ | 55.03 ↑ | 75.40 ↑ | 63.66 ↑ | 30.02 ↑ |
| | Oracle M3 | 71.72 ↑ | 55.80 ↑ | 77.20 ↑ | 62.44 ↓ | 29.57 ↓ |
| | SRLM M2 | 58.67 ↓ | 46.67 ↓ | 59.80 ↑ | 49.69 ↓ | 25.17 ↑ |
| | SRLM M3 | 46.55 ↓ | 34.47 ↓ | 49.20 ↓ | 48.06 ↓ | 22.14 ↓ |
| | CREAM M2 | 58.97 ↓ | 47.53 ↓ | 62.80 ↓ | 50.43 ↓ | 24.41 ↑ |
| | CREAM M3 | **62.08** ↑ | **48.81** ↑ | **64.60** ↑ | **51.22** ↑ | **25.85** ↑ |

oracle reward model (Oracle) can ensure high rewarding accuracy for annotations of self-generated preference data, thereby achieving the best performance overall. Notably, for Llama3, CREAM even outperforms Oracle except on SIQA dataset, showcasing the superior performance of CREAM. This superiority underlines the success of the proposed method in mitigating the rewarding bias issue. (4) The consistent performance improvements of CREAM across iterations validate the effectiveness of the proposed regularization method in mitigating the rewarding bias issue.

### 3.3 Analysis

#### 3.3.1 Analysis of Rewarding

**Rewarding Consistency.** We first examine the consistency of rewards of different methods using their corresponding models from the last iteration in Table 2. Here, we use the proposed Consistency Rate $\mathcal{C}$, Kendall correlation coefficient $\tau$, Spearman correlation coefficient, and TopOrder metrics to measure the consistency, where the TopOrder metric evaluates whether the final paired preference data remains the same, calculated as follows:

$$\text{TopOrder}_j = \mathbb{1}\left[\arg\min J_j = \arg\min K_j\right] \cdot \mathbb{1}\left[\arg\max J_j = \arg\max K_j\right],$$

where $J_j$ and $K_j$ are the rankings of the responses provided by current model and the last iteration's model, respectively. This metric assesses whether both the least preferred and most preferred responses are consistently ranked across iterations. The results confirm that SRLMs exhibit a rewarding consistency issue. In contrast, our method CREAM can keep the ranking consistency across iterations thanks to the explicit regularization in the training.

**Prompt Rewarding v.s. DPO Rewarding.** As aforementioned, 7B level LLMs struggle with generating accurate rewards when using LLM-as-a-Judge prompting due to their limited capacity. Both Table 5 and Figure 3 clearly show that the SRLM with prompt rewarding is not effective for smaller LLMs, as the performance starts to decrease at the first iteration (M1 → M2) when trained on the self-rewarded preference data. In contrast, the adopted DPO rewarding method can be more suitable for such small LLMs. This is primarily because DPO rewarding is intrinsically aligned with the model's learning objective.

Table 2: Ranking consistency of CREAM and SRLM on M2 and M3 using Llama3.

| Method | Iteration | Consistency $\mathcal{C}$↑ | Kendall $\tau$ ↑ | Spearman ↑ | TopOrder ↑ |
|---|---|---|---|---|---|
| SRLM | M2 | 0.39 ± 0.21 | -0.22 ± 0.41 | 0.36 ± 0.24 | 0.03 ± 0.18 |
| CREAM | M2 | 0.73 ± 0.18 | 0.46 ± 0.35 | 0.77 ± 0.19 | 0.19 ± 0.39 |
| SRLM | M3 | 0.46 ± 0.19 | -0.08 ± 0.38 | 0.50 ± 0.22 | 0.12 ± 0.33 |
| CREAM | M3 | 0.92 ± 0.09 | 0.84 ± 0.19 | 0.95 ± 0.07 | 0.59 ± 0.49 |

**Ranking Accuracy.** We present the ranking accuracy in Figure 3 to provide an intuitive comparison the performance of the rewarding performance across different methods. The results include the ranking accuracy on self-generated preference data and the RewardBench (Lambert et al.) dataset, both of which is formulated as a ranking task to predict the preferred one between two responses. We use the self-generated preference data obtained by self-rewarding with ground truth ranking labels, for testing the model's in-domain ranking performance. The RewardBench dataset is used to assess the generalizability of the models beyond the training domain. CREAM consistently achieves higher ranking accuracy than baseline methods, which promises more reliable preference data for training.

### 3.3.2 Reliability of Self-Consistency

The most straightforward way to enhance the rewarding and ranking accuracy is by incorporating external reward models, such as the SRLM variant "Oracle" used in our experiments. The theoretical analysis in Eq. (2.8) suggests that we can mitigate the rewarding bias issue by calculating the ranking consistency between current model and other available reward models. However, it is not always feasible to have access to such external reward models in practice, such as the self-rewarding scenario. Thus, we instead propose to use the last iteration's model as the reference reward model to measure the consistency of rewards. To test this approach, we fine-tune the same M1 model using CREAM with two different reference reward models: the rewarding function of Oracle and ours model from the last iteration. As shown in Table 3, using a strong reward model as the consistency model can bring better regularization effect, especially for Llama2. However, we find that the last iteration's model also provides a reasonably reliable consistency signal for Llama3.

Table 3: Comparison of CREAM with oracle reward model and last iteration's model.

| Method | Arc-E | Arc-C | OBQA | SIQA | GSM8K |
|---|---|---|---|---|---|
| Llama3 M1 | 86.78 | 80.14 | 86.40 | 69.50 | 78.39 |
| CREAM | 88.89 | 80.89 | 88.00 | 69.79 | 81.04 |
| CREAM + Oracle | 88.51 | 81.06 | 86.20 | 72.21 | 79.91 |
| Llama2 M1 | 60.44 | 48.46 | 63.20 | 50.77 | 23.88 |
| CREAM | 58.97 | 47.53 | 62.80 | 50.43 | 24.41 |
| CREAM + Oracle | 62.42 | 48.72 | 66.00 | 51.13 | 22.52 |

### 3.3.3 Consistency Measurement

Besides the adopted Kendall $\tau$ coefficient, other metrics can also be used to measure the consistency between two preference ranking lists, such as Spearman coefficient (Spearman, 1904) and the aforementioned TopOrder method. We conduct a comparison experiments of using different consistency measurement methods in Table 4. We can observe that: (1) All these measurements are effective with CREAM, indicating the generalization and applicability of our regularized training approach. (2) Kendall correlation coefficient generally yields higher scores across various datasets compared to Spearman and TopOrder methods.

Table 4: Performance of CREAM with different consistency measurements.

| Method | Arc-E | Arc-C | OBQA | SIQA | GSM8K |
|---|---|---|---|---|---|
| Spearman M2 | 86.95 | **82.00** | 85.40 | 70.05 | 78.77 |
| TopOrder M2 | 87.25 | 80.12 | 86.88 | **70.83** | 79.75 |
| Kendall M2 | **88.89** | 80.89 | **88.00** | 69.79 | **81.04** |
| Spearman M3 | 88.76 | 81.83 | 90.00 | 70.98 | 79.15 |
| TopOrder M3 | 88.51 | 80.37 | 87.40 | 71.03 | 79.76 |
| Kendall M3 | **89.52** | **83.36** | **90.20** | **72.06** | **81.73** |

## 4 Conclusion

In this paper, we first formulate a generalized iterative preference fine-tuning framework for self-rewarding language models (SRLMs), which is also applicable to other iterative preference training methods. Then, we highlight the rewarding bias that emerges from overconfident preference labeling, which is particularly problematic for smaller LLMs, such as those with 7B parameters. This rewarding bias results in the accumulation of noisy and unreliable preference data, harming the preference training and hindering alignment performance of LLMs. To address this issue, we proposed the **C**onsistency **R**egularized s**E**lf-Rewarding l**A**nguage **M**odel (CREAM), which leverages the consistency of rewards across different iterations as a regularization signal. This approach allows the model to learn more selectively, emphasizing reliable preference data and avoiding overconfidence in preference labeling. Our experimental results on various natural language benchmarks demonstrate the effectiveness of the proposed method in mitigating the rewarding bias issue and improving the performance of SRLMs. We believe that these findings can provide valuable insights for future research on self-improvement methods of LLM alignment.

## REFERENCES

Hervé Abdi. The kendall rank correlation coefficient. *Encyclopedia of measurement and statistics*, 2:508–510, 2007.

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

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences, 2023.

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

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. Benchmarking foundation models with language-model-as-an-examiner. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

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

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, and et al. Internlm2 technical report, 2024.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4299–4307, 2017.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and et al. The llama 3 herd of models, 2024.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Ramé, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. Direct language model alignment from online AI feedback. *CoRR*, abs/2402.04792, 2024a.

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

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2(4):5, 2024.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55:248:1–248:38, 2023.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling. *CoRR*, abs/2403.13787.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

A Ian McLeod. Kendall rank correlation and mann-kendall trend test. *R package Kendall*, 602:1–10, 2005.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. 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. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022b.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728, 2019.

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

C Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

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

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

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

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

Yiyang Zhou, Zhiyuan Fan, Dongjie Cheng, Sihan Yang, Zhaorun Chen, Chenhang Cui, Xiyao Wang, Yun Li, Linjun Zhang, and Huaxiu Yao. Calibrated self-rewarding vision language models. *arXiv preprint arXiv:2405.14622*, 2024.

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

Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5982–5991, 2019.

## A Related Works

This paper mainly focuses on mitigating the rewarding bias issue in self-rewarding language models (SRLMs) (Yuan et al., 2024), which is a type of self-improvement method for LLM alignment. In this section, we introduce the progresses in LLM alignment and discuss the SRLMs in detail.

**LLM Alignment.** Alignment lies at the core of LLM research and applications, aiming to ensure that LLMs adhere to human values and preferences. RLHF established the foundational alignment training paradigm (Leike et al., 2018; Ziegler et al., 2019; Ouyang et al., 2022a), where it leverages human preference feedback to train a reward model, and then use this reward model to guide the LLM via reinforcement learning algorithms (Schulman et al., 2017). Recent efforts have been made to develop direct alignment methods (Rafailov et al., 2023; Dong et al., 2023; Azar et al., 2023; Ethayarajh et al., 2024; Meng et al., 2024; Hong et al., 2024), in order to reduce the costs and complexity of RLHF and make it more efficient and accessible. Representatively, DPO (Rafailov et al., 2023) as a representative direct alignment method, optimizes the LLM with annotated preference pairs, eliminating the need of training an additional reward model. However, most RLHF and direct alignment methods heavily rely on human-annotated preference data, where the data collection commonly involves human to distinguish the "good" responses from the "bad" ones, which is time-consuming and labor-intensive (Ouyang et al., 2022a; Bai et al., 2022). Thus, synthesizing preference data with minimal human efforts has become a valuable research direction.

**Self-Rewarding Language Model.** SRLM (Yuan et al., 2024) has emerged as a promising approach to address the challenge of preference data synthesis in a self-improvement manner. This method leverages the LLM itself to act as both the policy model and the reward model. The policy model can generate response candidates for unlabeled prompts, while the reward model uses LLM-as-A-Judge (Zheng et al., 2023; Bai et al., 2023; Dubois et al., 2024) prompting to reward and rank these responses based on their quality. The ranked responses are then used as preference pairs to train the LLM via DPO (Rafailov et al., 2023). And this process can be iteratively repeated to improve the alignment performance without human intervention. However, having the same LLM serve as both the policy and reward model, without any regularization, presents challenges in guaranteeing accurate rewards. This can lead to accumulated bias and noisy preference data, which ultimately harms the training. Other similar self-improvement methods (Huang et al., 2022; Zelikman et al., 2022; Chen et al., 2024; Guo et al., 2024b; Zhou et al., 2024) often either use the ground truth response to avoid annotation bias, or introduce an additional reward model to reduce the noise in annotations. In contrast, our work neither requires labeled data nor relies on external LLMs. Instead, we propose to use the consistency of rewards to mitigate the rewarding bias in SRLMs.

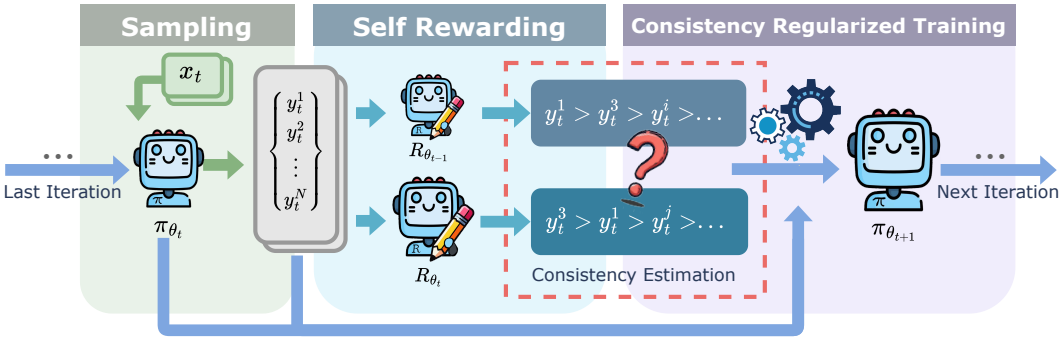


Figure 2: The flow of CREAM. In the response sampling stage, the policy model $\pi_{\boldsymbol{\theta}_t}$ generates $N$ responses. After that, CREAM uses the reward model $R_{\boldsymbol{\theta}_{t-1}}$ from the previous iteration to reward and rank these responses. Then, the rankings are compared with those generated by current reward model $R_{\boldsymbol{\theta}_t}$ to estimate the consistency rate. Finally, the policy model $\pi_{\boldsymbol{\theta}_t}$ is fine-tuned with consistency regularized preference training objective, resulting in the model $\pi_{\boldsymbol{\theta}_{t+1}}$ for next iteration.

Table 5: Results for SRLM with prompt rewarding method using Llama3. The dataset names are abbreviated.

| Dataset | M1 | M2 | M3 |
|---|---|---|---|
| Arc-E | 86.78 | 84.64↓ | 83.75↓ |
| Arc-C | 80.14 | 76.79↓ | 76.28↓ |
| OBQA | 86.40 | 80.40↓ | 80.20↓ |
| SIQA | 69.50 | 67.81↓ | 66.63↓ |
| GSM8K | 78.39 | 78.47↑ | 78.99↑ |

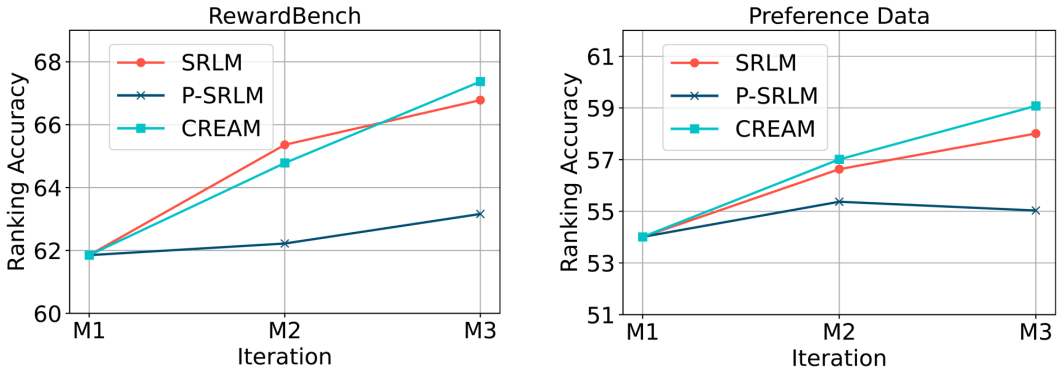


Figure 3: Ranking accuracy on RewardBench (left) and self-generated preference data (right). P-SRLM is SRLM with prompt rewarding.

## B Proof of the theorems and lemmas

### B.1 Proof of Theorem 2.1

*Proof of Theorem 2.1.* We denote iteration of the two-step algorithm as $t$. The algorithm starts from $(\boldsymbol{\theta}_t, z_t)$, and obtains $z_{t+1} = z_{\boldsymbol{\theta}_t}$ according to Eq. (2.3) in the preference-labeling step and then obtains $\boldsymbol{\theta}_{t+1}$ through the learning step. Since $z_{t+1} = \arg\min_z \mathcal{L}_{\text{DPO}}(\boldsymbol{\theta}_t; \mathbf{y}, \mathbf{y}', \mathbf{x}, z)$ for any $\mathbf{y}, \mathbf{y}', \mathbf{x}$ according to Eq. (2.3), we have that

$$\mathcal{L}(\boldsymbol{\theta}_t, z_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t, z_t). \tag{B.1}$$

And the learning step suggests that $\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, z_{t+1})$, yielding that

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}, z_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t, z_{t+1}). \tag{B.2}$$

Connecting Eq. (B.1) with Eq. (B.2) yields that the loss function $\mathcal{L}(\boldsymbol{\theta}, z)$ is monotonically decreasing, i.e.

$$\cdots \leq \mathcal{L}(\boldsymbol{\theta}_{t+1}, z_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t, z_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t, z_t) \leq \cdots. \tag{B.3}$$

Since $\mathcal{L}(\boldsymbol{\theta}, z)$ is upper bounded by 0, it suggests that the sequence of $\mathcal{L}(\boldsymbol{\theta}_t, z_t)$ will converge w.r.t. the growth of $t$. □

### B.2 Proof of Lemma 2.2

*Proof of Lemma 2.2.* We start by expanding $\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\mathcal{L}_{\text{reg}}(\boldsymbol{\theta};\mathbf{y},\mathbf{y}',\mathbf{x})$ as

$$\begin{aligned}\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\mathcal{L}_{\text{reg}}(\boldsymbol{\theta};\mathbf{y},\mathbf{y}',\mathbf{x}) &= \mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\log P_{\boldsymbol{\theta}}(z=1)+\log P_{\boldsymbol{\theta}}(z=0)\right]\\ &= \mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}),\mathbf{y}\prec\mathbf{y}'}\left[\log P_{\boldsymbol{\theta}}(z=1)+\log P_{\boldsymbol{\theta}}(z=0)\right]\\ &\quad+\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}),\mathbf{y}\succeq\mathbf{y}'}\left[\log P_{\boldsymbol{\theta}}(z=1)+\log P_{\boldsymbol{\theta}}(z=0)\right]\\ &= \mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}P_{\boldsymbol{\theta}_t}(z=0)\left[\log P_{\boldsymbol{\theta}}(z=1)+\log P_{\boldsymbol{\theta}}(z=0)\right]\\ &\quad+\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}P_{\boldsymbol{\theta}_t}(z=1)\left[\log P_{\boldsymbol{\theta}}(z=1)+\log P_{\boldsymbol{\theta}}(z=0)\right],\end{aligned} \tag{B.4}$$

where the second equation decompose the expectation $\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}$ into two expectation $\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}$ + $\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}$, the third equation extract the event $\mathbf{y}\succeq\mathbf{y}'$ as distribution $P_{\boldsymbol{\theta}_t}(z)$. Then Eq. (B.4) can be further written by

$$\begin{aligned}\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\mathcal{L}_{\text{reg}}(\boldsymbol{\theta};\mathbf{y},\mathbf{y}',\mathbf{x}) &= P_{\boldsymbol{\theta}_t}(z=1)\left[\log P_{\boldsymbol{\theta}}(z=1)+\log P_{\boldsymbol{\theta}}(z=0)\right]\\ &\quad+P_{\boldsymbol{\theta}_t}(z=0)\left[\log P_{\boldsymbol{\theta}}(z=1)+\log P_{\boldsymbol{\theta}}(z=0)\right],\end{aligned} \tag{B.5}$$

since both $\mathbf{y},\mathbf{y}'$ are generated from $\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})$, $P_{\boldsymbol{\theta}_t}(z=0)=P_{\boldsymbol{\theta}_t}(z=1)=0.5$. Thus Eq. (B.5) becomes

$$\mathbb{E}_{\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\mathcal{L}_{\text{reg}}(\boldsymbol{\theta};\mathbf{y},\mathbf{y}',\mathbf{x}) = 2\text{KL}(P_{\boldsymbol{\theta}}(z)\parallel P_{\boldsymbol{\theta}_t}(z)) = 2\text{KL}(P_{\boldsymbol{\theta}}(z)\parallel u(z)), \tag{B.6}$$

where $u(z)$ is the uniform binary distribution with $u(z=0)=u(z=1)=0.5$. □

### B.3 Proof Theorem 2.3

*Proof of Theorem 2.3.* We start by writing down each components in $\mathcal{L}(\boldsymbol{\theta},z)$ defined in Eq. (2.1) by

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta},z) &= \mathcal{L}_{\text{SFT}}(\boldsymbol{\theta};\mathcal{D}_{\text{S}})+\mathbb{E}_{\mathbf{x}\sim\mathcal{D}';\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}[\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta};\mathbf{y},\mathbf{y}',\mathbf{x},z)+\lambda\mathcal{L}_{\text{Reg}}(\boldsymbol{\theta};\mathbf{y},\mathbf{y}',\mathbf{x})]\\ &= \mathcal{L}_{\text{SFT}}(\boldsymbol{\theta};\mathcal{D}_{\text{S}})\\ &\quad+\mathbb{E}_{\mathbf{x}\sim\mathcal{D}';\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\Bigg[-z(\mathbf{y},\mathbf{y}',\mathbf{x})\log\sigma\left(\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}\right)-\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}\right)\right)\\ &\qquad-(1-z(\mathbf{y},\mathbf{y}',\mathbf{x}))\log\sigma\left(\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}\right)-\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}\right)\right)\\ &\qquad-\lambda\log\sigma\left(\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}\right)-\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}\right)\right)\\ &\qquad-\lambda\log\sigma\left(\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}\right)-\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}\right)\right)\Bigg]\\ &= \mathcal{L}_{\text{SFT}}(\boldsymbol{\theta};\mathcal{D}_{\text{S}})\\ &\quad+\mathbb{E}_{\mathbf{x}\sim\mathcal{D}';\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\Bigg[-(\lambda+z(\mathbf{y},\mathbf{y}',\mathbf{x}))\log\sigma\left(\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}\right)-\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}\right)\right)\\ &\qquad-(1+\lambda-z(\mathbf{y},\mathbf{y}',\mathbf{x}))\log\sigma\left(\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}\right)-\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}\right)\right)\Bigg]\end{aligned}$$

where the third equation absorbs the regularization into the DPO loss. Noticing that $\lambda+z(\mathbf{y},\mathbf{y}',\mathbf{x})+(1+\lambda-z(\mathbf{y},\mathbf{y}',\mathbf{x}))=1+2\lambda$, by dividing $(1+2\lambda)$ we have

$$\begin{aligned}\frac{\mathcal{L}(\boldsymbol{\theta},z)}{1+2\lambda} &= \frac{\mathcal{L}_{\text{SFT}}(\boldsymbol{\theta};\mathcal{D}_{\text{SFT}})}{1+2\lambda}\\ &\quad+\mathbb{E}_{\mathbf{x}\sim\mathcal{D}';\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\Bigg[-\frac{\lambda+z(\mathbf{y},\mathbf{y}',\mathbf{x})}{1+2\lambda}\log\sigma\left(\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}\right)-\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}\right)\right)\\ &\qquad-\frac{1+\lambda-z(\mathbf{y},\mathbf{y}',\mathbf{x})}{1+2\lambda}\log\sigma\left(\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}'|\mathbf{x})}\right)-\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}\right)\right)\Bigg].\end{aligned}$$

When $z(\mathbf{y},\mathbf{y}'\mathbf{x})=1$, $(\lambda+z(\mathbf{y},\mathbf{y}',\mathbf{x}))/(1+2\lambda)=1-\lambda/(1+2\lambda)$ and $(1+\lambda-z(\mathbf{y},\mathbf{y}',\mathbf{x}))(1+2\lambda)=\lambda/(1+2\lambda)$. Therefore, letting $\mathcal{C}_\lambda=\lambda/(1+2\lambda)$ yields that

$$\begin{aligned}\frac{\mathcal{L}(\boldsymbol{\theta},z)}{1+2\lambda} &= \frac{\mathcal{L}_{\text{SFT}}(\boldsymbol{\theta};\mathcal{D}_{\text{SFT}})}{1+2\lambda}\\ &\quad+\mathbb{E}_{\mathbf{x}\sim\mathcal{D}';\mathbf{y},\mathbf{y}'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}[(1-\mathcal{C}_\lambda)\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta};\mathbf{y},\mathbf{y}',\mathbf{x},z)+\mathcal{C}_\lambda\mathcal{L}_{\text{DPO}}(\boldsymbol{\theta};\mathbf{y},\mathbf{y}',\mathbf{x},1-z)],\end{aligned}$$

which completes the proof since minimizing $\mathcal{L}(\boldsymbol{\theta}, z)/(1+\lambda)$ is equivalent with minimizing $\mathcal{L}(\boldsymbol{\theta}, z)$ itself. □

### B.4 Proof of Lemma 2.4

*Proof of Lemma 2.4.* To begin with, according to the ranking of $J_{ij}$, the sufficient and necessary condition for $J_{ij} - J_{i'j} > 0$ is that $r_{ij} < r_{i'j}$. Similarly, the sufficient and necessary condition for $K_{ij} > K_{i'j}$ is that $r'_{ij} < r'_{i'j}$. As a result, the indicator becomes

$$\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) > 0] = \mathbb{1}[(r_{ij} - r_{i'j})(r'_{ij} - r'_{i'j}) > 0] \tag{B.7}$$

$$\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) < 0] = \mathbb{1}[(r_{ij} - r_{ij'})(r'_{ij} - r'_{i'j}) < 0]. \tag{B.8}$$

Since $r_{ij} > r_{i'j}$ yields $\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}$ under the input prompt $xb_j$ and language model $\boldsymbol{\theta}_t$, Eq. (B.7) becomes

$$\begin{aligned}\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) > 0] &= \mathbb{1}[r_{ij} > r_{i'j}]\,\mathbb{1}[r'_{ij} > r'_{i'j}] + \mathbb{1}[r_{ij} < r_{i'j}]\,\mathbb{1}[r'_{ij} < r'_{i'j}] \\ &= \mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}] \\ &\quad + \mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}].\end{aligned} \tag{B.9}$$

As a result, since $\mathbf{y}_{ij}$ are i.i.d. given $\mathbf{x}_j$, the expectation of first part of the Kendall's Tau coefficient is

$$\begin{aligned}&\mathbb{E}[\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) > 0]] - \mathbb{E}[\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) < 0]] \\ &= \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]] \\ &\quad + \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]] \\ &\quad - \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]] \\ &\quad - \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]] \\ &= \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t](\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}] - \mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}])] \\ &\quad + \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t](\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}] - \mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}])] \\ &= \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t](1 - 2\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}])] \\ &\quad - \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t](1 - 2\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}])] \\ &= \mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t] - \mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]] \\ &\quad + 2\mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}](\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t] - \mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t])] \\ &= 0 + 2\mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}](1 - 2\,\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t])]\end{aligned} \tag{B.10}$$

where the second equation merge the terms together, and the third equation is due to the fact $\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}] + \mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}] = 1$, the forth equation reorganize the term and the fifth equation is due to the fact that $\mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t] - \mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]] = 0$ due to symmetry. Similarly by reversing the $\prec$ and $\succ$, we can write Eq. (B.10) by

$$\begin{aligned}&\mathbb{E}[\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) > 0]] - \mathbb{E}[\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) < 0]] \\ &= 0 + 2\mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}](1 - 2\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t])].\end{aligned} \tag{B.11}$$

Adding Eq. (B.10) and Eq. (B.11) together yields

$$\begin{aligned}&2\mathbb{E}[\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) > 0]] - \mathbb{E}[\mathbb{1}[(J_{ij} - J_{i'j})(K_{ij} - K_{i'j}) < 0]] \\ &= 2\mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}] + \mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]] \\ &\quad - 4\mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]] \\ &\quad - 4\mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]] \\ &= 2 - 8\mathbb{E}_{\mathbf{y}_{ij}, \mathbf{y}_{i'j} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_j)}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]],\end{aligned} \tag{B.12}$$

where the final equation is because $\mathbb{E}[\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]] = \mathbb{E}[\mathbb{1}[\mathbf{y}_{ij} \succ \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_t]\,\mathbb{1}[\mathbf{y}_{ij} \prec \mathbf{y}_{i'j}|\mathbf{x}_j, \boldsymbol{\theta}_{t-1}]$ due to symmetry. Divide Eq. (B.12) by 2 yields the claimed result. □