Direct Reward Distillation: A Point-wise Alignment Approach

Anonymous ACL submission

Abstract

Direct Alignment Algorithms (DAAs) are widely used for aligning Large Language Models (LLMs) to human preferences. The current DAAs are using pairwise optimizing objectives based on the variants of Direct Preference Optimization (DPO). However, these methods only focus on the pairwise differences of the samples and cannot prevent optimization from reducing the probabilities of preferred responses. In this paper, we present Direct Reward Distillation (DRD), an algorithm that uses an explicit reward model to optimize the policy by setting an exact probability target for each response. DRD decouples target reward differentials and bias in aligning objectives and utilizing not only the relationship within response pairs but also the relationship among them. Our experiments show that DRD performs better than existing methods while providing controllability to the policy response probability.

1 Introduction

Large Language Model (LLM) alignment aims to enhance the ability of the model to align with human values and preferences, ensuring that it is helpful, honest, and harmless in serving humans (Ouyang et al., 2022). The typical LLM alignment approach, Reinforced Learning from Human Feedback (RLHF) (Ouyang et al., 2022), utilizes methods that rely on annotated preference data (i.e. positive and negative response pairs) to model human preferences through the Bradley-Terry (BT) model (Bradley and Terry, 1952). This approach first trains a reward model based on the preference data and then utilizes this model to guide the optimization of the LLM policy through online reinforcement learning techniques, such as Proximal Policy Optimization. Although RLHF has shown state-of-the-art performance so far, its pipeline is very complex, involving the training of multiple LLMs and sampling processes within the training loop. As a result, simpler alignment methods known as Direct Alignment Algorithms (DAAs) have gradually replaced RLHF as the mainstream approach (Gupta et al., 2025).

DAAs primarily incorporate Direct Preference Optimization (DPO) (Rafailov et al., 2024) and its various adaptations. DPO reparameterizes the reward function within the RLHF framework, suggesting that the optimizing policy can act as an implicit reward function. By optimizing the implicit reward function using the Bradley-Terry model, the policy aligns with preferences without the need to train an additional reward model or apply a reinforced learning process. As a result, DPO increases the generalization probability gap between the preferred responses and dispreferred ones.

Although DPO shares the same optimal objective and shows comparable performance with RLHF, it also has several proposed problems (Meng et al., 2024; Sharifnassab et al., 2024; Lin et al., 2024). Firstly, with a small β, DPO simultaneously reduces the probabilities of preferred responses and dispreferred responses, while increasing their gap (Meng et al., 2024; Hong et al., 2024). Although a larger probability gap indicates a more comprehensive alignment of preferences, making the probabilities of preferred responses too low can result in the LLM not being inclined to generate similar responses, further indicating a negative impact on policy (Gupta et al., 2025). Current approaches tend to solve this problem by adding different weights to the preferred and dispreferred responses in the training objective (Gupta et al., 2025; Hong et al., 2024). However, these methods break the objective consistency of DPO to RLHF. Moreover, the added hyperparameters require additional cost to locate the proper values in specific tasks.

Secondly, while dropping the phase of training an explicit reward model, the reward in DPO is calculated through a function involving the policy itself. Recent research points out that the implicit

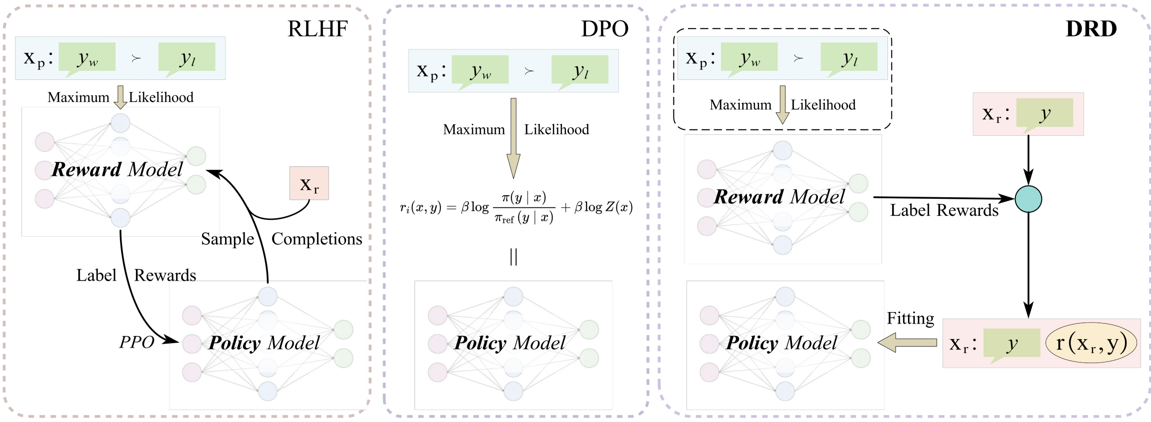


Figure 1: RLHF trains a reward model using the BT model and applies PPO for online optimizing the policy model. DPO uses the BT model to offline optimize the policy model. DRD uses a reward model (which may be trained by the BT model) to annotate the responses and offline distills the reward to the policy model.

reward model shows limited generalization capability (compared to explicit reward model training under the BT model in RLHF). The methods (Adler et al., 2024; Fisch et al., 2024) solve this problem by introducing an explicit reward model to the DPO and show an outperformance. They apply the rewards given by an explicit model to preference learning. Unlike the DPO's unbounded optimization, they set a target for the "reward gap" between each pair of responses and make the optimization more specific. However, these methods do not consider the drop in probability of preferred responses referred to above and they ignore the relationship among sample pairs indicated by given reward since they only take the reward differences between the responses within a pair.

In this paper, our aim is to answer the question: **Can DAA optimize the policy directly guiding the exact target of generation probability?** We observe that the problem of current DAAs reducing the preferred response probabilities is caused by their pairwise optimization structure whose adoption is due to the need to eliminate the normalization terms in the derivation of the RLHF objective for each sample (detailed in Section **??**). In this paper, we find that the terms can be derived from an invariant value and the optimal policy. By regarding this value as a hyperparameter, we propose Direct Reward Distillation (DRD), an algorithm using an explicit reward model to optimize the policy setting an exact target of probability for each response.

Compared to current DAAs, DRD solves the problem of reducing the probabilities of preferred responses. In fact, our method decouples target reward differentials and offsets of DAA and has controllability to the implicit reward value of the policy LLM. This provides practitioners with flexibility in adjusting optimization targets. In our experiments, we show that both the reward differentials and offsets affect the performance of the alignment process. Furthermore, our DRD utilizes the explicit reward model better (compared to previous works), referring not only to the relationship between the responses with the same prompt but also to the relationship among the responses with different prompts while preserving the simplicity of DPO. In particular, our DRD has no requirement for the reward model and how many responses each prompt has to participate in optimization. We present a standard way of training a typical BT reward model for DRD and utilize two responses for each prompt for training.

Our main contribution is Direct Reward Distillation (DRD), a pair-wise-optimization-free alignment algorithm with an explicit reward model which decouples the target reward differentials and bias and fully utilizes the reward information. Our experiments show that DRD is at least as effective as existing methods on the Ultra-Feedback (with Ultra-Chat) dataset, using language models Llama3-8B (Dubey et al., 2024), Qwen2.5-7B (Yang et al., 2024) and EuroLLM-9B (Martins et al., 2024).

2 Preliminaries

Given a large language model parameterized by θ, donated as π_θ. The current alignment algorithms aim to optimize π_θ by learning from annotated preference pairs.

RLHF: RLHF (Bai et al., 2022) fits a reward model to pairwise samples of human preferences and then uses Proximal Policy Optimization (PPO) to optimize a language model policy to produce responses that are assigned a higher reward without drifting excessively far from the original model. Consider an annotated dataset of pairwise samples $\mathcal{D}_p = \{x_i, y_w^i, y_l^i\}_{i=1}^N$, where x_i denotes the i^{th} prompt, y_w^i and y_l^i, respectively, represent the preferred and preferred responses to x_i. RLHF begins by modeling the probability of preferring y_w^i to y_l^i using the Bradley-Terry model (Bradley and Terry, 1952), which appoints the following probabilistic form:

$$p\left(y_w^i \succ y_l^i \mid x\right) = \sigma\left(r\left(x_i, y_w^i\right) - r\left(x, y_l^i\right)\right) \quad (1)$$

where σ represents the logistic function and $r(x_i, y_i)$ corresponds to a reward function r_ϕ (that is, the LLM classifier) that gives the estimation of y_i with respect to x_i according to human preference. Using maximum likelihood estimation to estimate the parameters of this function, we can optimize the classifier by the negative log-likelihood loss as below:

$$\begin{aligned}&\mathcal{L}_R\left(r_\phi, \mathcal{D}\right) = \\ &-\mathbb{E}_{\mathcal{D}}\left[\log\left(\sigma\left(r_\phi\left(x, y_w\right) - r_\phi\left(x, y_l\right)\right)\right)\right]\end{aligned} \quad (2)$$

The target model π_θ can then be trained by the feedback of the learned reward function. In general, we formulate the following optimization target for this learning process.

$$\max_{\pi_\theta} \mathbb{E}\left[r_\phi(x, y)\right] - \beta \mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \| \pi_{\mathrm{ref}}(y \mid x)\right] \quad (3)$$

where β is a parameter that controls the deviation of the target model π_θ from the status when training starts.

DPO: DPO (Rafailov et al., 2024) shows the possibility of keeping the same optimization target as RLHF without explicitly training a reward function and the implementation of RL. The loss function of DPO is presented below:

$$\begin{aligned}&\mathcal{L}_{\mathrm{DPO}}\left(\pi_\theta; \pi_{\mathrm{ref}}\right) = -\mathbb{E}_{\left(x, y_w, y_l\right) \sim D} \log \sigma \\ &\left(\beta \log \frac{\pi_\theta\left(y_w \mid x\right)}{\pi_{\mathrm{ref}}\left(y_w \mid x\right)} - \beta \log \frac{\pi_\theta\left(y_l \mid x\right)}{\pi_{\mathrm{ref}}\left(y_l \mid x\right)}\right)\end{aligned} \quad (4)$$

Notably, this optimization objective is based on a theoretical optimal π_θ beyond $r_U(x, y)$, which enables its equivalence with Eq.3.

3 Direct Reward Distillation

Aiming to guide the extract probability of responses for the policy LLM, we derived our training objectives from RLHF referring to previous works (Rafailov et al., 2024) and introduce a reward model to our DRD algorithm. By regarding the normalization term as a hyperparameter, DRD distills the reward of an explicit model to the implicit reward of policy LLM.

3.1 Reward Model

DRD uses the reward model to distill the rewards of an offline dataset to the policy LLM to guide the LLM to become the optimal policy under the objective Eq. 3. This ensure our DRD rely on a reward model with better generalization capability comparing to the DAAs without a reward model. Furthermore, our point-wise optimizing utilizes the reward relation between responses with different prompts rather than pair-wise DAAs.

Notably, DRD doesn't restrict to one specific reward model training method. In practice, for reward model training we follow the RLHF utilizing a Bradley-Terry model to model the preference of a pair-wise dataset (Rafailov et al., 2024). Specifically, we use the Eq. 2 to train a neural reward model which using a classifier processes the hidden state of the last token given by a pretrained LLM.

3.2 Direct Reward Distillation

Starting from the RLHF objective, we follow the previous work (Bai et al., 2022) and construct the reward function under the optimal solution $\hat{\pi}$ to Eq. 3 as follows:

$$r_i(x, y) = \beta \log \frac{\hat{\pi}(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z(x) \quad (5)$$

where $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$ represents the normalization term. Due to space limitation, we present a detailed deriving process in the Appendix A.1.

Algorithm 1: Direct Reward Distillation

Input: SFT model π_θ, Reward model r, Training Data $\mathcal{D}$, Norm Value Z_0, Training Epochs T, Learning Rate η

Output: Optimized Policy $\hat{\pi_\theta}$

$\pi_{ref} \longleftarrow \pi_\theta$;

foreach *Epoch t=1, 2, ..., T* **do**

  Get a batch of samples $\mathcal{D}_\mathcal{T} \subset \mathcal{D}$;

  $\mathcal{L}_\mathcal{T} \longleftarrow 0$;

  foreach $\left(x_T, y_T^1, y_T^2, ...\right) \in \mathcal{D}_\mathcal{T}$ **do**

    $Z_T = \frac{\pi_\theta(x_T - t_0|t_0}{\pi_{ref}(x_T - t_0|t_0)} Z_0$;

    Detach Z_T ;

    foreach y_T^i **do**

      $r_T \longleftarrow$
      $\beta \log \frac{\pi_\theta(y_T^i|x_T)}{\pi_{\text{ref}}(y_T^i|x_T)} + \beta \log Z_T$

      $\mathcal{L}_\mathcal{T} \longleftarrow$
      $\mathcal{L}_\mathcal{T} + \left(r(x_T, y_T^i) - r_T\right)^2$;

  $\pi_\theta \longleftarrow \pi_\theta - \eta \nabla \left(\frac{\mathcal{L}_\mathcal{T}}{|\text{the number of y in } \mathcal{D}_\mathcal{T}|}\right)$;

$\hat{\pi_\theta} \longleftarrow \pi_\theta$;

The normalization term $Z(x)$ changes with prompts x, resulting in the result that the implicit reward target needs exact $\mathbf{Z} = \{Z(x_1), Z(x_2), Z(x_N)\}$. Considering that the reward model partition of x and y doesn't effect the given reward in Eq. 5, we can deriving a relationship between $Z(x, y)$ and its prefix $Z(x)$ as below:

$$\frac{Z(x, y)}{Z(x)} = \frac{\hat{\pi}(y \mid x)}{\pi_{ref}(y \mid x)} \tag{6}$$

Through this relationship, we can assume an imaginary overall prefix t_0 which fits to every prompt x_i. Thus the normalization term $Z_0 = Z(t_0)$ whose defination is $Z_0 = \sum_y \pi_{\text{ref}}(y \mid t_0) \exp\left(\frac{1}{\beta} r(t_0, y)\right)$. This indicates that the relationships among $\mathbf{Z}$ are related to the $\hat{\pi}$ and π_{ref}. Once obtaining the value of $Z(x_i)$, our DRD optimize the policy utilizing the MSE Loss:

$$\mathcal{L}_{DRD}\left(\pi_\theta, r, \mathbf{Z}; \mathcal{D}\right) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\left(r(x, y) - \beta \log \frac{\pi(y \mid x)}{\pi_{\text{ref}}(y \mid x)} - \beta \log Z(x)\right)^2\right] \tag{7}$$

3.3 Optimization

DRD distills the explicit reward to improve the LLM policy. Referring to the work of (Adler et al., 2024), we adopt the phase of including more than one response per prompt for training to ensure better preference supervision. Notably, while the assumption of Z_0 requires an overall prefix t_0 which every prompt x_i has, DRD theoretically restricts the prompts to have the same "start token". It is easy to meet this condition since almost every LLM template stipulates the first token (e.g., "⟨|im_start|⟩" or "User").

Theorem 3.1. *Suppose a reward model $r(x, y)$ gives a reward to the dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, infinite various $r(x, y)$ can be constructed ensuring: 1. $r(x_i, y_i) = r'(x_i, y_i) for x_i, y_i \in \mathcal{D}$. 2. For all x_i, y_i, x_j, y_j in the language space, $[r(x_i, y_i) - r(x_j, y_j)][r'(x_i, y_i) - r'(x_j, y_j)] > 0$*

The actual value of Z_0 is calculated by its definition. However, in DRD, we regard it as a hyperparameter. As our derivation in App. **??** proving Thm. 3.1, there're different reward models having different Z_0 act the same in the optimization. We approximate $\hat{\pi}$ to π_θ in optimization, ensuring the consistency of the optimal solution of DRD. The experiments further confirm that this approximation does not compromise the convergence.

We use a pseudocode presented as Algorithm 1 to show the DRD optimization. DRD aims to optimize the implicit reward of the policy and treats the normalization term Z_i as a constant. After obtaining the target of $\log \hat{\pi}(y \mid x)$, DRD utilizes an MSE loss for training referring to previous work (Fisch et al., 2024).

3.4 The Interpretation of DRD

Our DRD utilizes Eq. 6 to generate an approximate normalized term to Eq. 5 and uses the MSE loss for optimization. While combining $Z(t_0)$ to Eq. 5 using Eq. 6, we can result to the below equation:

$$r(x, y) = \beta \log \frac{\hat{\pi}(t_0 \mid x, y - t_0)}{\pi_{\text{ref}}(t_0 \mid x, y - t_0)} + \beta \log Z(t_0) \tag{8}$$

Which is the Eq. 5 in a certain situation. In particular, in Algo. 1, Z_i does not contribute to the gradient since the generation probabilities of the prompts are within our optimization scope, which makes DRD optimization different than the direct utilization of Eq. 8.

As Eq. 8 shares the same optimal policy with DRD, we can infer from it that β presents the level of reward differences of our optimization target. The smaller β is, the greater the gap among our reward target which is the same in the work of (Rafailov et al., 2024). Z_0 in DRD presents an "offset" to the rewards. While Z_0 grows down, all the reward targets move upwards. This ensures that DRD controls the generation probabilities from simultaneous decreases.

4 Experiments

We experiment with our DRD based on the below pretrained LLMs: Llama3-8B (Dubey et al., 2024), Qwen2.5-7B (Bai et al., 2023) and EuroLLM-9B (Martins et al., 2024). In this section, our aim is to present the advantages of our DRD versus other direct alignment baselines. We start from the base models and fine-tuning them to gain the instruction-following capability. Reward models are trained on a pairwise preference data set. Then we use the reward models to annotate the rewards of this preference dataset and use DRD to optimize the fine-tuned LLMs. Notably, we keep sampling two responses each prompt in order to keep the scale of training data is same to DRD and all baselines.

4.1 Datasets and Evaluations

We follow the typical training pipeline of Zephyr (Tunstall et al., 2023) and SimPO (Meng et al., 2024) to select datasets. For the supervised fine-tuning phase, we apply the UltraChat-200k dataset (Ding et al., 2023) to train our base models. Notably, we optimize the base models utilizing the multi-turn dialogue templates of their chat derivatives. For reward model training and alignment optimization, we apply the UltraFeedback dataset (Cui et al., 2023). This approach provides a high level of reproducing. Below we give their brief introductions:

•UltraChat-200k is a multi-turn instructional conversation dataset that contains 207,865 conversations for training. UltraChat-200k is designed by a principle that attempts to capture the breadth of interactions that a human might have with an AI assistant and then employs meta-information, in-context expansion, and iterative prompting to scale up the number of instructions. The constructors use LLMs only to generate the conversations.

•UltraFeedback is a large-scale, high-quality, and diversified AI feedback dataset, which contains over 1 million GPT-4 feedback for user-assistant conversations from various aspects. It is constructed beyond a compiled diverse array of over 60,000 instructions and 17 models from multiple sources and then utilizes GPT-4 for annotation with a bunch of techniques to alleviate annotation biases and improve feedback quality to the greatest extent. Notably, we utilize binary preferences from UltraFeedback by selecting the highest mean score as the preferred response and one of the remaining three at random as dispreferred referring to (Tunstall et al., 2023). The total number of data pairs for training is 61,135.

Table 1: The reward model training results.

Model Setting	Small		Large	
	Loss	Acc	Loss	Acc
RM-Base	0.0621	0.975	0.0539	0.982
RM-SFT	0.0463	0.979	0.035	0.988
DPO-Implicit	0.2039	0.9521	0.2463	0.9660

For evaluation benchmarks, we apply the widely used benchmarks for general instruction-following capability: Alpaca-Eval2 (Dubois et al., 2024) and MT-Bench (Zheng et al., 2024). These benchmarks evaluate the LLM's versatile conversational capabilities utilizing different queries. Alpaca-Eval2 constructs its 805 queries from 5 datasets and MT-Bench contains 80 queries sampled from 8 different categories. Both benchmarks rely on a LLM-as-judge evaluating methods. Notably, we use GPT-4 (Achiam et al., 2023) as the annotator for them. For Alpaca-Eval2, we present the results of win rate (WR) and length-controlled win rate which reflects the evaluation results eliminating the effect of model verbosity over a base response which is sampled from GPT-4 Turbo (Achiam et al., 2023). For MT-Bench, we report the average overall score calculated based on the judgment of GPT-4.

4.2 Baselines

We compare our DRD with different direct alignment algorithm baselines. Except the widely used and introduced **DPO**, **SLiC-HF** (Zhao et al., 2023) using linear ranking losses for optimization instead of the BT model. **IPO** (Azar et al., 2024), constructed a general preference learning structure objective deriving from which DPO is a special case, bypasses the BT modelization assumption for preferences, and utilizes an MSE loss. **ORPO** (Hong et al., 2024) drop the reference model in DPO and introduce an odd ratio to directly optimize the probabilities of the policy model while jointly training with an objective of preferred response maximum likelihood loss. **SimPO** (Meng et al., 2024) uses the average log probability of a sequence as the implicit reward and introduces a target reward margin in the DPO objective. Robust Preference Optimization (RPO) (Fisch et al., 2024) introduces an explicit reward model to distill the reward gaps to the policy model. Notably, we use the same reward model to provide the reward gaps as our DRD uses. We only use one reward model in RPO to ensure the fairness of our DRD and RPO. Notably, except IPO, all the above methods do not share the same optimal solution consistency as DPO and DRD to RLHF.

4.3 Implement Details

We present our detailed Implement Details in the App. B

4.4 Reward Model

Our DRD doesn't specify the approach of the reward model used to give the reward. Here we present a demonstrative reward model training process. We utilize the Bradley-Terry model to train an explicit reward model that gives a reward score through a randomly initialized classifier on the hidden state of the last token of a pretrained model's output. To compare the performances of explicit reward models initialized with the base model and the SFT model and the implicit reward model indicated in Eq. 5, we utilize all the preference pairs in UltraFeedback (regarded as "large" setting) or 10000 pairs randomly sampled from it (regarded as "small" setting) either to train the reward models based on Llama3. Taking the loss of training ends and the metrics of reward accuracy (i.e. the accuracy of the reward model gives a larger reward to preferred response than dispreferred ones) on the test set of UltraFeedback, we present the results in Tab. 1.

We can observe that the explicit reward model initialized by the SFT model performs best among the three. The either explicit model shows an apparent advantage to the implicit model. This indicates the benefits of using an explicit reward model for alignment as our DRD. Following the results, we train the reward model of Qwen2.5 and EuroLLM using their SFT model instead of directly using the base model.

4.5 Main Results

The main results of our experiments are presented in Tab. 2. Remarkably, while all the direct alignment baselines optimize the SFT model to a better conversational capability referring to the benchmarks, DRD outperforms all the baselines in all settings except SimPO on EuroLLM-9B on the Alpaca-Eval 2 win-rate metric. This illustrates the advantages of DRD compared to current alignment methods. Notably, DRD achieves an 82.83% increase over the SFT model and a 5.04% increase over RPO who performs best among the baselines in the Alpaca-Eval 2 win rate metric based on Llama3-8B and this advantage comes to 73.47% and 12.31% on the length-controlled win rate. For Qwen2.5-7B, DRD gains 14.79% and 14.98% advantages compared to the best baseline on win rate and length-controlled win rate of Alpaca-Eval 2. For EuroLLM-9B, DRD gains a 6.29% advantage on the length-controlled win rate.

A cursory examination reveals that our DRD has an obvious outperformance over all the direct alignment baselines across all tasks. Such a pattern underscores the effectiveness of DRD in improving LLM's ability in preference learning. DRD not only introduces an explicit reward model that has a better generalization capability to the alignment training but also provides a more stable training target using point-wise loss and prevents the continual decreasing of preferred response probabilities.

4.6 Analysis

We here present a detailed analysis of our DRD controls and the alignment process of the Policy. As shown in Fig. 2, we conclude:

•DRD utilizes point-wise loss to optimize the policy model. It set a target to the chosen reward of the policy model thus we can observe from Fig. 2(a) that the reward of both chosen and rejected rewards are symmetrically separated from each other while keeping a clear stable mean value. This mean value is the Z_0 value set to be stable in the training process. While Z_0 grows larger, this mean value drops.

•From another perspective, the effect of Z_0 and β in DRD is more clearer in Fig. 2(b). While Z_0 grows larger, the chosen reward of the training end decreases. While β grows smaller, this decreasing trend becomes slower. It can be inferred that when Z_0 is enough larger, the chosen reward can be smaller than utilizing DPO.

•As for the gap between chosen rewards and rejected rewards in the training ends, β can have a significant effect. While β drops, this gap grows rapidly. One of our DRD's main effectiveness is decoupling the reward gap and the mean value of

Table 2: Overall result.

Methods	Llama3-8B			Qwen2.5-7B			EuroLLM-9B		
	AlpacaEval 2		MT-Bench	AlpacaEval 2		MT-Bench	AlpacaEval 2		MT-Bench
	WR(%)	LC(%)		WR(%)	LC(%)		WR(%)	LC(%)	
SFT	3.35	5.82	5.0	5.41	10.69	5.7	4.11	7.81	5.3
SLiC-HF	9.87	11.06	5.5	8.55	12.86	6.0	8.28	9.03	5.4
DPO	18.32	17.63	6.5	18.12	23.16	**6.8**	12.52	16.02	6.0
IPO	14.92	15.24	6.1	13.25	14.47	6.4	11.38	11.98	5.8
ORPO	11.97	13.535	5.7	9.10	12.72	6.2	9.29	12.26	5.8
SimPO	18.42	19.97	**6.6**	17.32	23.28	6.7	**14.92**	16.53	**6.2**
RPO	18.52	19.24	**6.6**	17.74	22.14	6.6	14.24	14.59	6.1
DRD	**19.51**	**21.94**	**6.6**	**20.82**	**26.04**	**6.8**	14.11	**17.64**	**6.2**

Table 3: Overall result.

Methods	MMLU	GSM8K	ARC-Easy	ARC-Hard	MathQA	SocialQA	Avg.
SFT	63.81	25.84	52.82	48.29	26.73	50.25	44.62
SLiC-HF	64.76	28.32	65.00	50.94	26.37	53.73	48.19
DPO	64.88	28.84	49.37	39.25	**28.88**	37.45	41.45
IPO	63.25	28.96	60.29	45.30	27.03	40.78	44.27
ORPO	65.02	26.24	63.95	49.82	24.14	53.69	47.14
SimPO	63.47	25.02	44.57	36.6	25.42	36.83	38.65
DRD	**64.93**	**31.72**	**69.49**	**55.38**	27.19	**54.95**	**50.61**

the alignment target. It can be seen in Fig. 2(c) that Z_0 does little effectiveness to the reward gap.

•From Fig. 2(d) we can observe that the performance of the alignment algorithm is affected by the compound of other factors. Neither reward gap nor the chosen reward can reflect the final performance independently.

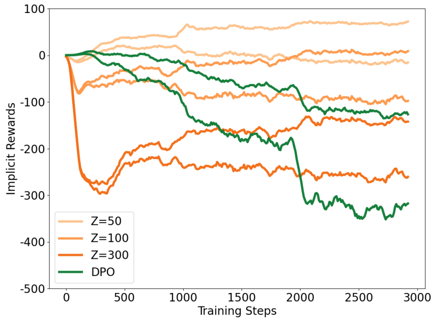
4.7 Downstream Tasks Evaluation

To examine how exactly the models perform in different fields, we evaluate all the models reported in Tab. 2 which is based on Llama3-8B to various downstream tasks. Specifically, we include the MMLU (Hendrycks et al., 2020), GSM8K (Cobbe et al., 2021), ARC-Easy and Challenge (Clark et al., 2018), MathQA (Amini et al., 2019), and SocialQA (Sap et al., 2019). As reported in (Meng et al., 2024), several direct alignment algorithms may drop the model performances in reasoning tasks. Thus we mainly choose the reasoning tasks in our evaluation and the widely used MMLU. Notably, except MMLU, all the tasks are evaluated through the CoT Pass@1 zero-shot setting. We set the sampling temperature to 0.0 as adopt the greedy sampling method.

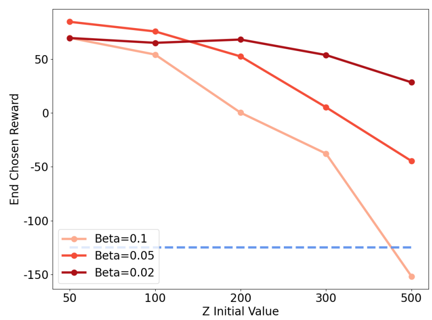
The results are presented in Tab. 3. We can observe that DRD performs better to all the baselines. While alignment methods as DPO and SimPO obviously drop the model's reasoning capabilities, DRD does not decrease the ability of SFT model and instead improves the reasoning ability of the model through alignment. We infer that some baselines dropping the model's reasoning capability may caused by the significant decrease of preferred response probabilities the alignment methods do to the policy model. While "heavily" optimizing the model to align with human preference, the training process overfits the model and weakens its generalization ability. This proves the advantages of DRD.

5 Related Works

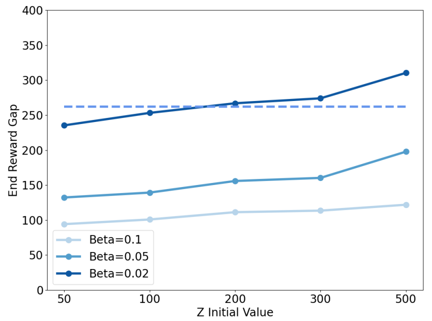
Large language models (LLMs) have shown great zero-shot and few-shot performance (Brown et al., 2020; Chowdhery et al., 2023; Radford et al., 2019). After being pretrained on a large corpus, LLMs obtain the ability to complete downstream tasks, following the supervised fine-tuning instructions and human-written responses (Chung et al., 2024; Mishra et al., 2021; Sanh et al., 2021). Despite the success of instruction tuning, preference optimization has shown great effectiveness in aligning LLMs with humans (Bai et al., 2022). As reinforcement Learning with Human Feedback (RLHF) (Bai et al., 2022) is a complex and often unstable procedure (Pal et al., 2024), DPO (Rafailov et al., 2024) has been proposed as a simple and computationally



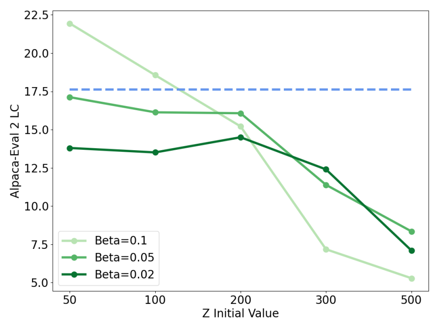
(a) Training Rewards.



(b) Chosen rewards.



(c) Reward gap.



(d) Performances.

Figure 2: Analysis of DRD training process. The analysis experiments are conducted on Llama3-8B under different hyperparameters. The blue dashed line represents the performance of DPO.

lightweight method with no need for additional reward function training. Specifically, it derives the optimal policy of RLHF objective and reparameters the reward model using the current policy (i.e. using policy as an implicit reward model). Through this way, the optimization to policy model transfers to the optimization of the reparametered reward function using BT model.

Various works have been proposed based on the DPO method for better performances. ORPO (Hong et al., 2024) and SimPO (Meng et al., 2024) focus on regularization of sequence length aiming to reduce the phenomenon that DPO tend to increase the response length of policy LLM. DPOP (Pal et al., 2024), KTO (Ethayarajh et al., 2024) reduce the problem of DPO by lowering the preferred response probabilities by increasing the weight of the preferred term in the training objective. However, these methods break the theoretical basis of DPO and obtain uncertain gains. In particular, Robust Preference Optimization (Fisch et al., 2024) and Reward-Aware Preference Optimization (Adler et al., 2024) introduce an explicit general reward model to provide a target reward difference for each prompt. However, they still adopt the pairwise optimization method which cannot prevent the chosen reward decrease problem and overlook the relationship among samples given by the explicit reward model.

Our DRD proposes a point-wise direct alignment method that has better utilization of the reward model information and strengthened control over optimization objectives.

6 Conclusion

In this paper, we propose a Direct Reward Distillation (DRD) method that utilizes a point-wise target for aligning the model.

Compared to the existing direct alignment approaches that are based on pair-wise losses to optimize the policy model. DRD prevents the policy model from dropping the generation probability of the preferred responses and referring not only to the relationship between the responses with the same prompt but also to the relationship among the responses with different prompts.

Experimental results on various reasoning tasks and datasets demonstrate the superior performance of our DRD which consistently outperforms a wide range of baseline approaches.

7 Limitations

Our paper presents a simple and effective method to align the LLMs to human performances. We present our experiments based on a typical trained Bradley-Terry model using exactly the same data used for alignment optimization. It would be better to discuss more about the reward models and do a more comprehensive experiment about the number of responses for each prompt used in the optimization as DRD doesn't restrict to the pairwise training structure.

8 Discussion of Ethical Considerations

Our proposed methods are used to improve the capabilities of LLMs. Using DRD training LLMs may cause an environmental impact as all other training methods do.

For the permissions of our used artifact, each of our used models (Llama3-8B, Qwen2.5-7B, EuroLLM-9B) and the datasets (UltraChat, UltraFeedBack, GSM8K, ARC, MathQA) are open-sourced and can be found from Github or Huggingface. Secondly, all the models can not be used commercially.

We utilize all the models and datasets consistent with their intended use. We do not provide extra data. Our construction of self-training data using the LLMs presents the answers to the datasets, which is the purpose LLMs are designed.

The datasets we used contain no information that names or uniquely identifies individual people or offensive content.

References

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

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. 2024. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

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.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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.

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

Yezeng Chen, Zui Chen, and Yi Zhou. 2024a. Brain-inspired two-stage approach: Enhancing mathematical reasoning by imitating human thought processes. *arXiv preprint arXiv:2403.00800*.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language

A Deriving the optimal solution of RLHF

A.1 Proof for optimal solution of RLHF

We construct our proof following the previous works[1, 2]. From Eq. **??**, our optimizing target is:

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}[r(x,y)] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi(y\mid x)\|\pi_{\mathrm{ref}}(y\mid x)\right] \tag{1}$$

Notably, we can derive as:

$$\begin{aligned}
&\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}[r(x,y)] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi(y\mid x)\|\pi_{\mathrm{ref}}(y\mid x)\right] \\
&\qquad = \max_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[r(x,y) - \beta\log\frac{\pi(y\mid x)}{\pi_{\mathrm{ref}}(y\mid x)}\right] \\
&\qquad = \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y\mid x)}{\pi_{\mathrm{ref}}(y\mid x)} - \frac{1}{\beta}r(x,y)\right] \\
&\qquad = \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y\mid x)}{\frac{1}{Z(x)}\pi_{\mathrm{ref}}(y\mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)} - \log Z(x)\right]
\end{aligned} \tag{2}$$

where we define as :

$$Z(x) = \sum_{y}\pi_{\mathrm{ref}}\left(y\mid x\right)\exp\left(\frac{1}{\beta}r(x,y)\right) \tag{3}$$

Notably, $Z(x)$ is a function of only x and π_{ref}. We can additionally define:

$$\hat{\pi}(y\mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y\mid x)\exp\left(\frac{1}{\beta}r(x,y)\right) \tag{4}$$

As is a probability distribution which holds $\sum_y \pi^*(y\mid x) = 1$. Using the $Z(x)$, we can re-organize the Eq. 1 as:

$$\begin{aligned}
&\min_{\pi}\mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y\mid x)}{\hat{\pi}(y\mid x)}\right] - \log Z(x)\right] = \\
&\quad \min_{\pi}\mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\mathrm{KL}}\left(\pi(y\mid x)\|\hat{\pi}(y\mid x)\right) - \log Z(x)\right]
\end{aligned} \tag{5}$$

Since $Z(x)$does not depend on π, the optimal solution is achieved by the policy that minimizes the first term. The KL divergence is minimized in the situation where two distributions are equal. Thus we have the optimal solution:

$$\pi(y\mid x) = \hat{\pi}(y\mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y\mid x)\exp\left(\frac{1}{\beta}r(x,y)\right) \tag{6}$$

B Implement Details

The experiments are carried out on 16 A100-80G GPUs with a Linux system. For all baselines and DRD, we search the hyperparameters as we present the details in the Appendix C. For the SFT phase, we train 2 epochs in each setting and report the performance of the best checkpoint. For the alignment phase, we train 3 epochs and take the same approach. We use *Pytorch*[1] and *Huggingface*[2] as tools for

[1]https://pytorch.org/
[2]https://huggingface.co/

the implementation. For alignment, we apply experiments based on *trl*[3]. All the generations during the evaluation process were done using *vllm* [3][4]. The code will be released on GitHub[5].

C HyperParameter Search

Table 1: Hyperparameter search range.

Methods	Search Range
DPO	$\beta \in [0.05, 0.1, 0.5, 1.0]$ $lr \in [1e-7, 2e-7, 5e-7, 1e-6]$
SLiC-HF	$\lambda \in [0.05, 0.1, 0.5, 1.0, 5, 0]$ $lr \in [1e-7, 2e-7, 5e-7]$
IPO	$\beta \in [0.05, 0.1, 0.5, 1.0]$ $lr \in [1e-7, 2e-7, 5e-7, 1e-6]$ $\alpha \in [0.25, 0.5, 1, 2]$
ORPO	$\tau \in [0.01, 0.05, 0.1, 1.0]$
SimPO	$\beta \in [1.0, 2.0, 2.5]$ $\gamma \in [0.3, 0.5, 0.7, 1.0, 1.5]$
RPO	$\beta \in [0.05, 0.1, 0.5, 1.0]$
DRD	$\beta \in [0.05, 0.1, 0.5, 1.0]$ $lr \in [1e-7, 2e-7, 5e-7, 1e-6]$ $Z_0 \in [-50, 500]$

Notably, we are referring to the papers [2, 4, 5, 6, 7] to set the search ranges.

[3]https://github.com/huggingface/trl
[4]https://github.com/vllm-project/vllm
[5]http://github.com/xxxxxx