In-Dataset Trajectory Return Regularization for Offline Preference-based Reinforcement Learning

Songjun Tu[1,2,3], **Jingbo Sun**[1,2,3], **Qichao Zhang**[1,3*], **Yaocheng Zhang**[1,3], **Jia Liu**[4], **Ke Chen**[2], **Dongbin Zhao**[1,2,3]

[1]State Key Laboratory of Multimodal Artificial Intelligence Systems, CASIA, Beijing, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[4]School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China
{tusongjun2023,zhangqichao2014}@ia.ac.cn

Abstract

Offline preference-based reinforcement learning (PbRL) typically operates in two phases: first, use human preferences to learn a reward model and annotate rewards for a reward-free offline dataset; second, learn a policy by optimizing the learned reward via offline RL. However, accurately modeling step-wise rewards from trajectory-level preference feedback presents inherent challenges. The reward bias introduced, particularly the overestimation of predicted rewards, leads to optimistic trajectory stitching, which undermines the pessimism mechanism critical to the offline RL phase. To address this challenge, we propose In-Dataset Trajectory Return Regularization (DTR) for offline PbRL, which leverages conditional sequence modeling to mitigate the risk of learning inaccurate trajectory stitching under reward bias. Specifically, DTR employs Decision Transformer and TD-Learning to strike a balance between maintaining fidelity to the behavior policy with high in-dataset trajectory returns and selecting optimal actions based on high reward labels. Additionally, we introduce an ensemble normalization technique that effectively integrates multiple reward models, balancing the trade-off between reward differentiation and accuracy. Empirical evaluations on various benchmarks demonstrate the superiority of DTR over other state-of-the-art baselines.

Extended version — https://arxiv.org/html/2412.09104

Introduction

Designing complex artificial rewards in reinforcement learning (RL) is challenging and time-consuming (Skalse et al. 2022; Wang et al. 2024a). Preference-based reinforcement learning (PbRL) addresses this by leveraging human feedback to guide policies, demonstrating success in aligning large language models (Ouyang et al. 2022) and robot control (Liang et al. 2022). Recently, considering the growing utilization of offline data in aiding policy optimization via offline RL (Fang et al. 2022; Chen, Li, and Zhao 2024), offline PbRL (Shin, Dragan, and Brown 2023) has gained attention. This approach involves training a reward model with limited human feedback, annotating rewards for a reward-free offline dataset, and applying offline RL to learn policies.

Despite significant advancements in offline PbRL, learning an accurate step-wise reward model from trajectory-wise preference feedback remains inherently challenging due to limited feedback data (Zhang et al. 2023), credit assignment (Kim et al. 2023) and neural network approximation errors (Zhu, Jordan, and Jiao 2023). The introduced reward bias adds potential brittleness to the pipeline, leading to suboptimal performance (Yu et al. 2022; Hu et al. 2023). To mitigate this issue, some studies have aimed to enhance the robustness of the reward model (Shin, Dragan, and Brown 2023; Gao et al. 2024), yet ignoring the potential influence of reward bias in offline RL. Alternatively, approaches that bypass reward modeling and directly optimize policy using preference (An et al. 2023; Hejna et al. 2024) struggle to achieve out-of-distribution (OOD) generalization, limiting their ability to outperform the dataset (Xu et al. 2024).

For the policy learning, most of offline PbRL methods (Christiano et al. 2017; Yuan et al. 2024; Gao et al. 2024) apply the learned reward function directly to downstream TD-Learning (TDL) based offline RL algorithms, such as CQL (Kumar et al. 2020) and IQL (Kostrikov, Nair, and Levine 2022). However, these TDL-based methods do not account for potential bias in the predicted rewards. The introduced reward bias, especially overestimated rewards, can lead to optimistic trajectory stitching and undermine the pessimism towards OOD state-action pairs in offline TDL algorithms (Yu et al. 2022). To minimize the impact of reward bias, it is necessary to consider being pessimistic about overestimated rewards during the offline policy learning phase (Zhan et al. 2024). Apart from TDL-based offline algorithms, another methodology, conditional sequence modeling (CSM) (Emmons et al. 2022) such as Decision Transformer (DT) (Chen et al. 2021), has not yet been investigated in offline PbRL. This type of imitation-based approach learns a maximum return policy with in-dataset trajectories by assigning appropriate trajectory reweighting. Compared to TDL, which extracts policy based on value functions, CSM extracts policy conditioned on in-dataset trajectory returns, thereby potentially constraints its trajectory stitching capability (Yamagata, Khalil, and Santos-Rodriguez 2023). Although a limited trajectory stitching ability may not be anticipated for offline RL with ground-truth (GT) rewards, this limitation can be considered as a pessimistic safeguard to alleviate the

*Corresponding author.

inaccurate trajectory stitching due to incorrect reward labels and maintain fidelity to the behavior policy with high in-dataset trajectory returns in offline PbRL. These properties trigger our further thought:

Is it possible to leverage the trajectory return-based CSM to mitigate inaccurate stitching of TDL for offline PbRL?

Building upon these insights, we propose In-**D**ataset **T**rajectory Return **R**egularization **(DTR)** for offline PbRL, which integrates CSM and TDL to achieve the offline RL policy based on in-dataset trajectory returns and annotated step-wise preference rewards. Specifically, DTR consists of the following three core components: (1) a DT (Chen et al. 2021) structure based policy is employed to associate the return-to-go (RTG) token with in-dataset individual trajectories, aintaining fidelity to the behavior policy with high trajectory-wise returns; (2) a TDL module aims to utilize Q function to select optimal actions with high step-wise rewards and balance the limited trajectory stitching ability of DT; and (3) an ensemble normalization that effectively integrates multiple reward models to balance reward differentiation and inaccuracy.

By combining CSM and TDL dynamically, DTR mitigates the potential risk of reward bias in offline PbRL, leading to enhanced performance. Our contributions are summarized below:

- We propose DTR, a valid integration of CSM with DT-based regularization to address the impact of TD-learning stitching caused by reward bias in offline PbRL.
- We introduce a dynamic coefficient in the policy loss to balance the conservatism and exploration, and an ensemble normalization method in reward labeling to balance reward differentiation and inaccuracy.
- We prove that extracting policies from in-dataset trajectories leads to provable suboptimal bounds, resulting in enhanced performance in offline PbRL.
- Our experiments on public datasets demonstrate the superior performance and significant potential of DTR.

Related Works

Offline PbRL. To enhance the performance of offline PbRL, some works emphasize the importance of improving the robustness of reward model, such as improving the credit assignment of returns (Early et al. 2022; Kim et al. 2023; Verma and Metcalf 2024), utilizing data augmentation to augment the generalization of reward model (Hu et al. 2024b). Other approaches modify the Bradley-Terry model to optimize preferences directly and avoid reward modeling, such as DPPO (An et al. 2023) and CPL (Hejna et al. 2024). A related SOTA work is FTB (Zhang et al. 2023), which optimizes policies based on diffusion model and augmented trajectories without TDL. In contrast, our approach balances trajectory-wise DT and step-wise TD3 (Fujimoto, Hoof, and Meger 2018) to mitigate the risk of reward bias and leads to enhanced performance.

In theory, (Zhu, Jordan, and Jiao 2023) proves that the widely used maximum likelihood estimator (MLE) converges under the Bradley-Terry model in offline PbRL with the restriction to linear reward function. (Hu et al. 2023) stresses that pessimism about overestimated rewards should be considered in offline RL. (Zhan et al. 2024) relaxes the assumption of linear reward and provides theoretical guarantees to general function approximation. Based on these results, we further provide theoretical suboptimal bound guarantees for offline PbRL under the estimated value function.

Improving the Stitching Ability of CSM. Due to the transformer architecture and the paradigm of supervised learning, CSM has limited trajectory stitching ability. Therefore, some works aggregate the input of the transformer-based policy or modify its structure to adapt to the characteristics of multimodal trajectories (Shang et al. 2022; Zeng et al. 2024; Kim et al. 2024). The alternative approach leverages Q value to enhance trajectory stitching capability such as CGDT (Wang et al. 2024b), QDT (Yamagata, Khalil, and Santos-Rodriguez 2023), Q-Transformer (Chebotar et al. 2023) and recent QT (Hu et al. 2024a) for offline RL with GT rewards.

Preliminaries

Learning Rewards From Human Feedback

Following previous studies (Lee, Smith, and Abbeel 2021; Kim et al. 2023), we consider trajectories of length H composed of states and actions, defined as $\sigma = \{s_k, a_k, \ldots, s_{k+H}, a_{k+H}\}$. The goal is to align human preference y between pairs of trajectory segments σ^0 and σ^1, where y denotes a distribution indicating human preference, captured as $y \in \{1, 0, 0.5\}$. The preference label $y = 1$ indicates that σ^0 is preferred to σ^1, namely, $\sigma^0 \succ \sigma^1$, $y = 0$ indicates $\sigma^1 \succ \sigma^0$, and $y = 0.5$ indicates equal preference for both. The preference datasets are stored as triples, denoted as $\mathcal{D}_{pref}$: (σ^0, σ^1, y).

The Bradley-Terry model (Bradley and Terry 1952) is frequently employed to couple preferences with rewards. The preference predictor is defined as follows:

$$P_\psi[\sigma^1 \succ \sigma^0] = \frac{\exp\left(\sum_t \hat{r}_\psi(s_t^1, a_t^1)\right)}{\sum_{i\in\{0,1\}} \exp\left(\sum_t \hat{r}_\psi(s_t^i, a_t^i)\right)} \quad (1)$$

where $\hat{r}_\psi$ is the reward model to be trained, and ψ is its parameters. Subsequently, the reward function is optimized using the cross-entropy loss, incorporating the human ground-truth label y and the preference predictor P_ψ:

$$\mathcal{L}_{\text{CE}} = -\mathbb{E}_{(\sigma^0,\sigma^1,y)\sim\mathcal{D}_{pref}}\Big\{(1-y)\log P_\psi[\sigma^0 \succ \sigma^1] + y \log P_\psi[\sigma^1 \succ \sigma^0]\Big\} \quad (2)$$

In the offline PbRL, we assume there exists a small dataset $\mathcal{D}_{pref}$ with preference labels along with a much larger unlabeled dataset $\mathcal{D}$ without rewards or preference labels. We label the offline dataset $\mathcal{D}$ with estimated step-wise rewards $\hat{r}$. Then we can obtain the trajectory dataset by calculating trajectory-wise RTG $\hat{R}$ for individual trajectories in $\mathcal{D}$. The re-labeled dataset is used for downstream offline RL. The notation "in-dataset trajectory" indicates that the trajectory is in the offline dataset $\mathcal{D}$.

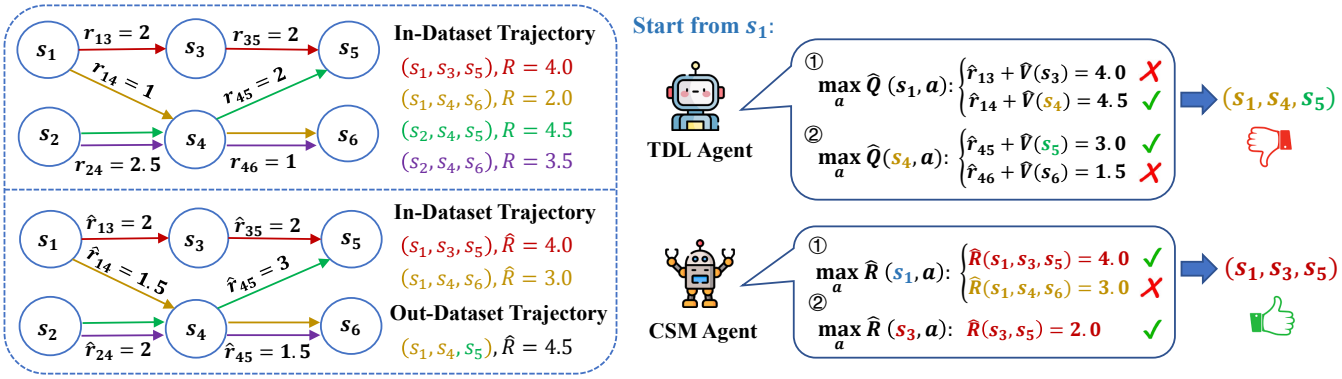


Figure 1: A deterministic MDP that illustrates the potential failure mode of offline PbRL. Since the state transition is deterministic, we represent the trajectory as a sequence of states. The four colored lines represent four trajectories in the preference dataset $\mathcal{D}_{pref}$. The reward r indicates the GT reward, and $\hat{r}$ indicates the estimated reward from preference. Starting at state s_1, inaccurate estimated rewards result in a greater return of the stitched trajectory (s_1, s_4, s_5) than the in-dataset trajectory (s_1, s_3, s_5) for TDL agent, which is factually incompatible. In contrast, CSM agent gets correct actions that align closely with the behavior policy conditioned on high in-dataset trajectory return.

Return-Based Conditional Sequence Modeling

RL is formulated as a Markov Decision Process (MDP) (Sutton 2018). A MDP is characterized by the tuple $M = \langle S, A, P, r, \gamma \rangle$, where S is the state space, A is the action space, $P : S \times A \times S \to \mathbb{R}$ is the transition probability distribution, $r : S \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. The objective of RL is to determine an optimal policy π that maximizes the expected cumulative reward: $\pi = \arg\max_\pi \mathbb{E}_{s_0, a_0, \ldots} [\sum_{t=0}^{\infty} \gamma^t r(s_t)]$.

Return-based CSM addresses sequential decision problems through an autoregressive generative model, thus avoiding value estimation (Brandfonbrener et al. 2022). A prominent example is DT (Chen et al. 2021), which considers a sequence of trajectories of length H: $\tau_t = (R_{t-H+1}, s_{t-H+1}, a_{t-H+1}, \cdots, R_t, s_t, a_t)$, where R_t denotes return-to-go (RTG): $R_t = \sum_{t'=t}^{T} r_{t'}$. During the training phase, the supervised loss between the target policy and behavior policy is computed using mean squared error (MSE) loss. During the inference phase, DT rolls out the estimated action $\hat{a}_t$ with the provided RTG $\hat{R}_t$. Although CSM harnesses the power of large models for supervised learning, its limited trajectory stitching capability constrains the potential for improvement from suboptimal data.

Methods

In this section, we first introduce the problem of reward bias in offline PbRL using a toy example and analyze the TDL and CSM methods in such scenarios to further emphasize our motivation. Then, we provide a detailed pipeline and the overall framework for the DTR method. Finally, we prove that extracting policies with in-dataset trajectory return regularization leads to provable suboptimal bounds.

Rethinking Stitching in PbRL: A Toy Example

We use the deterministic MDP depicted in Figure 1 as an example. In this MDP, the agent starts from s_1 or s_2, transitions through s_3 or s_4, and finally reaches s_5 or s_6. The GT reward for each state transition is denoted as r. We estimate stepwise rewards $\hat{r}$ from trajectory-level pairwise preferences. Typically, the preference dataset $\mathcal{D}_{pref}$ is gathered through a limited number of queries, making it difficult to cover all possible trajectories in the offline dataset $\mathcal{D}$. Suppose $\mathcal{D}_{pref}$ includes pairwise comparison labels for four trajectories and their intermediate segments:

$$\mathcal{D}_{pref} = \begin{pmatrix} (s_1, s_3, s_5)_{red}, & (s_1, s_4, s_6)_{yellow}, & y = 1 \\ (s_2, s_4, s_5)_{green}, & (s_1, s_4, s_6)_{yellow}, & y = 1 \\ (s_2, s_4, s_5)_{green}, & (s_1, s_3, s_5)_{red}, & y = 1 \\ (s_1, s_3, s_5)_{red}, & (s_2, s_4, s_6)_{purple}, & y = 1 \end{pmatrix}$$

Here we use $(\cdot)_{color}$ to denote the trajectory and its color depicted in Figure 1. Suppose the trajectories in $\mathcal{D}$ are identical to those in $\mathcal{D}_{pref}$. There are three possible trajectories beginning from s_1: (s_1, s_3, s_5), (s_1, s_4, s_6) and (s_1, s_4, s_5). The first two are intrinsic in $\mathcal{D}$ (in-dataset), while the last one is formed by stitching together two trajectory fragments (out-dataset). The trajectory with the highest GT return is (s_1, s_3, s_5). However, due to the reward bias, we assume $\hat{r}_{45} = 3$ is higher than the ground-truth $r_{45} = 2$. Consequently, the estimated return of the stitched trajectory (s_1, s_4, s_5) is higher than that of (s_1, s_3, s_5). Such failures of estimated rewards are common when using a small-scale $\mathcal{D}_{pref}$ to label a large-scale $\mathcal{D}$.

We now scrutinize the performance of CSM and TDL methods under the reward bias. Starting from s_1, TDL chooses the stitched trajectory (s_1, s_4, s_5) that maximizes the expected cumulative reward. In contrast, CSM identifies the in-dataset trajectory that maximizes the RTG for s_1, considering only trajectories within $\mathcal{D}$ that contain s_1, specifically (s_1, s_3, s_5) and (s_1, s_4, s_6). As a result, CSM opts for s_3 over s_4, leading to s_5. To put it simply, CSM prefers conservative choices to learn behaviors under limited preference queries, seeking optimal actions that align closely with the behavior policy conditioned on RTG. Conversely, TDL may

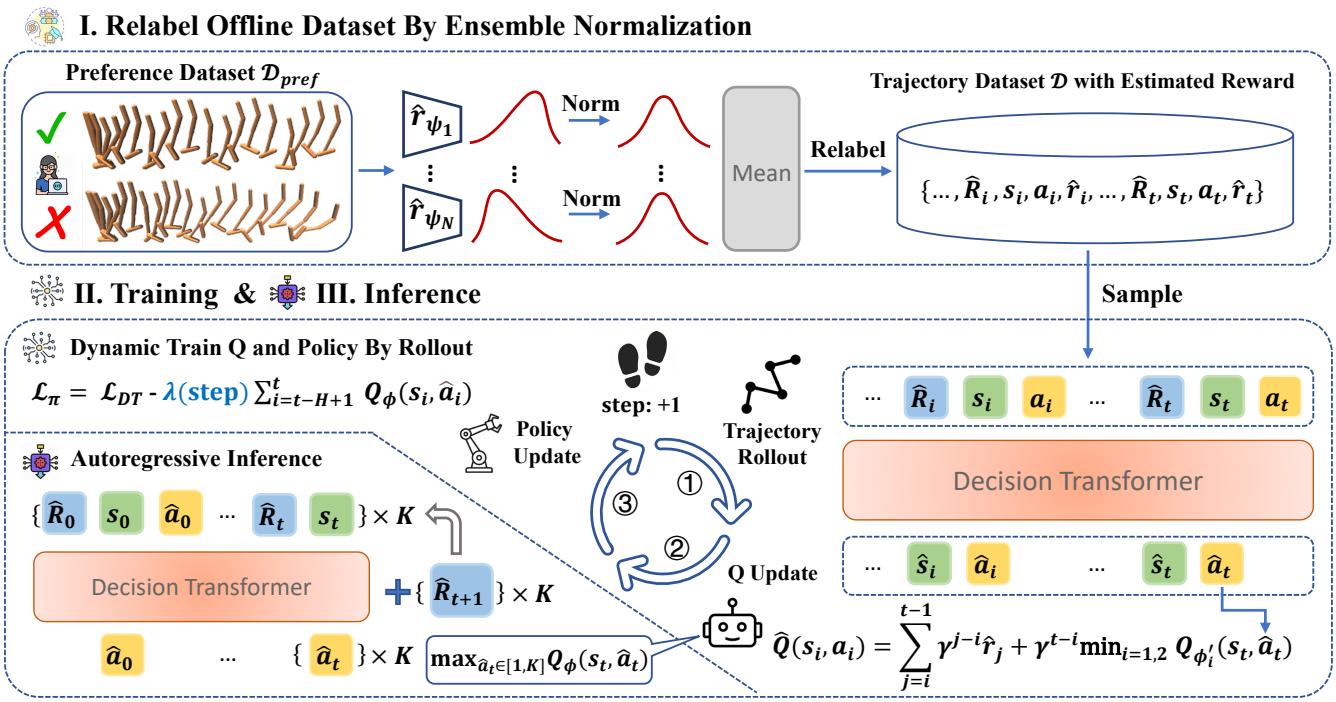


Figure 2: The overall framework DTR. Phase I: Utilize preference data $\mathcal{D}_{pref}$ to learn the reward model, then label the offline dataset $\mathcal{D}$ by ensemble normalization. Phase II: Train the Q function and policy network, comprising three components: trajectory rollout, Q update, and policy update. Phase III: Autoregressively infer actions based on the target RTG and select the action with the highest Q value.

fail due to optimistic trajectory stitching under reward bias. Therefore, we hope to balance conservatism and exploration by integrating CSM and TDL dynamically for offline PbRL.

In-dataset Regularization: Training and Inference

In the offline policy training phase, we aim to utilize the DT and TD3 to achieve a balance between maintaining fidelity to the behavior policy with high in-dataset trajectory returns and selecting optimal actions with high reward labels. Specifically, we sample a mini-batch trajectories from $\mathcal{D}$, and use DT as the policy network to rollout the estimated actions $\{\hat{a}_i\}_{i=t-H+1}^{t}$. Due to the powerful capability of autoregressive generative model, we also rollout estimated states $\{\hat{s}_i\}_{i=t-H+1}^{t}$ to strengthen policy representation (Liu et al. 2024). Considering the trajectory $\tau_t = \{\cdots, \hat{R}_i, s_i, a_i, \cdots, \hat{R}_t, s_t, a_t\}$, the self-supervised loss of DT can be expressed as:

$$\mathcal{L}_{DT} = \mathbb{E}_{\tau_t \sim \mathcal{D}} \sum_{i=t-H+1}^{t} \underbrace{\|\hat{s}_i - s_i\|^2}_{\text{Auxiliary Loss}} + \underbrace{\|\hat{a}_i - a_i\|^2}_{\text{Goal-BC Loss}} \quad (3)$$

Additionally, we train the Q function to select optimal actions with high reward labels. A natural idea is to use the estimated trajectory sequence for n-step TD bootstrapping. Following (Fujimoto, Hoof, and Meger 2018) and (Hu et al. 2024a), the Q target and Q loss are defined as follows:

$$\hat{Q}(s_i, a_i) = \sum_{j=i}^{t-1} \gamma^{j-i} \hat{r}_j + \gamma^{t-i} \min_{i=1,2} Q_{\phi_i}(s_t, \hat{a}_t) \quad (4)$$

$$\mathcal{L}_Q = \mathbb{E}_{\tau_t \sim \mathcal{D}} \sum_{i=t-H+1}^{t} \left\| Q_\phi(s_i, a_i) - \hat{Q}(s_i, a_i) \right\|^2 \quad (5)$$

To prevent failures from incorrect trajectory stitching, as highlighted in the toy example, we dynamically integrate the policy gradient to train the policy network. Meanwhile, in Theorem 1, we will show that extracting policies from in-dataset trajectories leads to provable suboptimal bounds. In the early training stage, the policy loss $\mathcal{L}_\pi$ primarily consists of supervised loss $\mathcal{L}_{DT}$ to ensure that the policy learns the trajectories of in-dataset optimal return and the corresponding Q function. Subsequently, we gradually increase the proportion of policy gradient to facilitate trajectory stitching near the optimal range within the distribution. In summary, we minimize the following policy loss:

$$\mathcal{L}_\pi = \mathcal{L}_{DT} - \lambda(\text{step})\, \mathbb{E}_{\tau_t \sim \mathcal{D}} \sum_{i=t-H+1}^{t} Q_\phi(s_i, \hat{a}_i) \quad (6)$$

where $\lambda(\text{step})$ considers the normalized Q value and increases linearly with the training steps:

$$\lambda(\text{step}) = \frac{\eta}{\mathbb{E}_{(s,a)\sim\tau_t} |Q_\phi(s, a)|}, \ \eta = \frac{\text{step} \times \eta_{max}}{\text{max\_step}} \quad (7)$$

Different from TD3BC (Fujimoto and Gu 2021) with the constant η, our approach uses progressively increasing coefficients to maintain training stability. This follows a straightforward principle: "Learn to walk before you can run."

After the training, we get a trained policy π_θ and Q network Q_ϕ. During the inference stage, we autoregressively rollout the action sequence almost following the setting of DT (Chen et al. 2021). The difference is that we cannot update the RTG with the rewards provided by the online environment. An alternative way is to replace the environment reward with a learned reward model:

$$\hat{R}_{t+1} = \hat{R}_t - \hat{r}_t \tag{8}$$

However, the reward model has limited generalization for OOD state-action pairs, and the additional model leads to greater consumption of computing resources. Therefore, we propose a simple calculation: subtract a fixed value from the next timestep's RTG $\hat{R}_{t+1}$ until it is reduced to 0 at the end of the episode:

$$\hat{R}_{t+1} = \hat{R}_t - \hat{R}_0/\text{max\_timestep} \tag{9}$$

The reason for this simplification is that we normalize the rewards so that candidate states with high values tend to have similar rewards at each step. In implementation, we calculate K initial RTGs as targets: $\{\hat{R}_0^i\}_{i=1}^k = \{0.5, 0.75, 1, 1.5, 2\} \times \text{Return}_{\max}$, here we have $K = 5$, and $\text{Return}_{\max}$ is the maximum return in $\mathcal{D}$. Through the parallel reasoning of Transformer, K actions are derived, and the one with the highest Q value is executed.

Relabel Offline Data by Ensemble Normalization

During the annotation reward phase, we introduce a simple but effective normalization method to highlight reward differentiation, and further improve the performance of downstream policies. Specifically, we train N-ensemble MLP reward models $\{\hat{r}_{\psi_i}\}_{i=1}^N$ from the offline preference dataset $\mathcal{D}_{pref}$ with the loss function defined in Equation 2, then annotate the offline data $\mathcal{D}$ with predicted rewards.

We observe that the trained reward model labels different state-action pairs with only minor differences (as detailed in the Experiments Section and Figure 4a), and these indistinct reward signals may lead to exploration difficulties and low sample efficiency (Jin et al. 2020). While direct reward normalization can amplify these differences, it also increases uncertainty, resulting in inaccurate value estimation and high variance performance (Gao et al. 2024).

To balance these two aspects, we propose ensemble normalization, which first normalizes the estimates of each ensemble and then averages them:

$$\hat{r}_\psi = \text{Mean}(\text{Norm}(\hat{r}_{\psi_1}), \cdots, \text{Norm}(\hat{r}_{\psi_N})) \tag{10}$$

Notably, ensemble normalization is a plug-and-play module that can enhance reward estimation without any modifications to reward training.

Theoretical Analysis

Suppose the preference dataset $\mathcal{D}_{pref}$ with N_p trajectories of length H_p and the reward-free offline dataset $\mathcal{D}$ with N_o trajectories of length H_o. Then, consider the following general offline PbRL algorithm:

1. Construct the Reward Confidence Set:

First, estimate the parameters $\widehat{\psi} \in \mathbb{R}^d$ of reward model via MLE with Equation 2. Then, construct a confidence set $\Psi(\zeta)$ by selecting reward models that nearly maximize the log-likelihood of $\mathcal{D}_{pref}$ to a slackness parameter ζ.

2. Bi-Level Optimization of Reward and Policy:

Identify the policy $\widehat{\pi}$ that maximizes the estimated policy value $\widehat{V}$ under the least favorable reward model $\widetilde{\psi}$ with both $\mathcal{D}_{pref}$ and $\mathcal{D}$:

$$\widetilde{\psi} = \arg\min_{\psi \in \Psi(\zeta)} \widehat{V}_\psi - \mathbb{E}_{\tau \sim \mu_{ref}}[r_\psi(\tau)], \;\; \widehat{\pi} = \arg\max_\pi \widehat{V}_{\widetilde{\psi}}$$

And we have the following theorem:

Theorem 1 (Informal) *Suppose that: (1) the dataset $\mathcal{D}_{pref}$ and $\mathcal{D}$ have positive coverage coefficients $C_p^\dagger$ and $C_o^\dagger$; (2) the underlying MDP is a linear MDP; (3) the GT reward $\psi^\star \in \mathcal{G}_\psi$; (4) $0 \le r_\psi(\tau) \le r_{\max}$, and $\|\psi\|_2^2 \le d$ for all $\psi \in \mathcal{G}_\psi$ and $\tau \in \mathcal{T}$. and (5) $\widehat{\pi}$ is any mesurable function of the data $\mathcal{D}_{pref}$. Then with probability $1 - 2\delta$, the performance bound of the policy $\widehat{\pi}$ satisfies for all $s \in \mathcal{S}$,*

$$\begin{aligned} \textit{SubOpt}(\widehat{\pi}; s) \le & \sqrt{\frac{cC_\psi^2(\mathcal{G}_\psi, \pi^\star, \mu_{ref})\kappa^2 \log(\mathcal{N}_{\mathcal{G}_\psi}/N_p\delta)}{N_p}} \\ & + \frac{2cr_{\max}}{(1-\gamma)^2}\sqrt{\frac{d^3\xi_\delta}{N_pH_pC_p^\dagger + N_oH_oC_o^\dagger}} \end{aligned} \tag{11}$$

where $\xi_\delta = \log(4d(N_pH_p + N_oH_o)/(1-\gamma)\delta)$, and other variables involved are not related to with N_p or N_o. For detailed definitions, see Appendix A.

Remark 1 *The theorem extends offline RL theory specifically to offline PbRL, leading to a theoretical upper bound for offline PbRL. In order to guarantee this bound, the learned policy $\widehat{\pi}$ is any measurable function of $\mathcal{D}_{pref}$ should be satisfied. In other words, if $\widehat{\pi} \in \mathcal{D}_{pref}$, the assumption naturally holds. A relaxed condition is $\widehat{\pi} \in \mathcal{D}$, since trajectories in $\mathcal{D}_{pref}$ are often sampled from $\mathcal{D}$. This theoretical analysis guides us to make more use of in-dataset trajectories for policy optimization to ensure marginal performance improvements. Accordingly, our DTR method utilizes CSM for optimizing in-dataset trajectories to establish reliable performance bound, while employing TDL to enhance the utilization of out-dataset trajectories. We elaborate on this finding with experiments in Appendix E7.*

Experiments

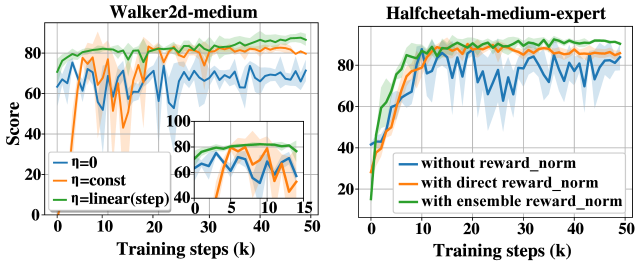
In this section, we evaluate DTR and other baselines on various benchmarks. We aim to address the following questions: (1) Can DTR mitigate the risk of reward bias and lead to enhanced performance in offline PbRL? (2) What specific roles do the proposed modules play in the proposed DTR? (3) Can DTR still perform well with small amounts of preference feedback?

Dataset	Pb-IQL	Pb-TD3BC	DPPO	OPRL	FTB	Pb-DT*	Pb-QT*	DTR (Ours)*	DTR (Best)*
Walker2D-m	78.4	26.3	28.4	80.8	79.7	71.4 ± 5.1	80.1 ± 1.4	**86.6** ± 2.8	88.3 ± 2.7
Walker2D-m-r	67.3	47.2	50.9	63.2	79.9	51.1 ± 10.5	79.3 ± 1.5	**80.8** ± 2.8	84.4 ± 2.1
Walker2D-m-e	109.4	74.5	108.6	109.6	109.1	108.0 ± 0.2	109.5 ± 0.6	**109.7** ± 0.3	111.1 ± 0.3
Hopper-m	50.8	48.0	44.0	59.8	61.9	49.8 ± 4.8	81.3 ± 8.8	**90.7** ± 0.6	94.5 ± 0.4
Hopper-m-r	87.1	25.8	73.2	72.8	90.8	67.0 ± 8.2	84.5 ± 12.8	**92.5** ± 0.9	96.0 ± 1.6
Hopper-m-e	94.3	97.4	107.2	81.4	110.0	**111.3** ± 0.1	109.4 ± 1.8	109.5 ± 2.4	112.3 ± 0.3
Halfcheetah-m	43.3	34.8	38.5	**47.5**	35.1	42.5 ± 0.8	42.7 ± 0.3	43.6 ± 0.3	44.2 ± 0.3
Halfcheetah-m-r	38.0	38.9	40.8	**42.3**	39.0	37.6 ± 0.6	39.9 ± 0.7	40.6 ± 0.2	41.6 ± 0.2
Halfcheetah-m-e	91.0	73.8	**92.6**	87.7	91.3	68.7 ± 9.1	78.2 ± 11.6	91.9 ± 0.4	93.5 ± 0.3
Average	73.29	51.86	64.91	71.68	77.42	67.49	78.32	**82.88**	85.10

Table 1: The performance comparison of DTR and different baselines on Gym-MuJoCo locomotion. In the first column, -m, -m-r, and -m-e are abbreviations for the medium, medium-replay, and medium-expert datasets, respectively. The results of Pb-IQL and Pb-TD3BC are from Uni-RLHF benchmark (Yuan et al. 2024), the results of OPRL and FTB are from the experimental section of FTB (Zhang et al. 2023), the -m-r and -m-e results of DPPO are from (An et al. 2023), and the -m results are reproduced by ourselves. For the remaining methods with *, we record the average normalized score of the 10 rollouts of the last checkpoint and run the experiment under 5 random seeds, and finally record the mean score and ± denotes the standard deviation. The **bold font** indicates the algorithm with the best performance, and the underline indicates the second one.

Dataset	Pb-IQL	Pb-DT	Pb-QT	DTR (Ours)
Pen-h	99.8 ± 8.3	**115.3** ± 2.0	111.2 ± 3.9	114.0 ± 9.6
Pen-c	93.7 ± 14.1	**104.6** ± 13.4	69.1 ± 12.0	86.6 ± 6.4
Door-h	9.7 ± 1.4	14.2 ± 1.4	26.0 ± 5.0	**33.3** ± 8.6
Door-c	2.2 ± 0.6	7.7 ± 0.9	10.7 ± 1.9	**12.6** ± 0.6
Hammer-h	11.8 ± 2.0	2.0 ± 0.3	15.0 ± 5.7	**21.5** ± 6.9
Hammer-c	11.4 ± 2.7	4.0 ± 2.1	14.6 ± 1.5	**26.7** ± 10.3
Average	38.10	41.29	41.10	**49.11**

Table 2: The performance comparison on Adroit manipulation platform. In the first column, -h and -c are abbreviations for the human and clone datasets.



(a) Ablation study on η. (b) Ablation study on norm_r.

Figure 3: Training score curve for the ablation study.

Setup. We select three tasks from the Gym-MuJoCo locomotion suite (Brockman et al. 2016), and three tasks from the Adroit manipulation platform (Kumar 2016). We utilize the Uni-RLHF dataset (Yuan et al. 2024) as preference dataset $\mathcal{D}_{pref}$, which provides pairwise preference labels from crowdsourced human annotators. The offline dataset $\mathcal{D}$ is sourced from the D4RL benchmark (Fu et al. 2020).

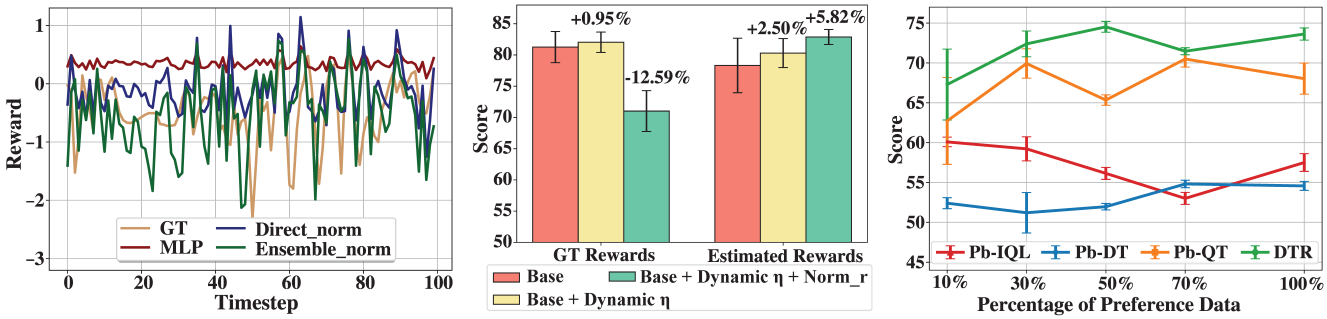
Baselines. (1) Pb (Kim et al. 2023) including PT-IQL and Pb-TD3BC, which perform IQL or TD3BC with preference-based trained reward functions, respectively; (2) DPPO (An et al. 2023), which designs a novel policy scoring metric and optimize policies without reward modeling; (3) OPRL (Shin, Dragan, and Brown 2023), which performs IQL with ensemble-diversified reward functions; (4) FTB (Zhang et al. 2023), which generates better trajectories with higher preferences based on diffusion model. For a more intuitive comparison, we propose several variants: (5) Pb-DT and Pb-QT, which slightly modify CSM models DT (Chen et al. 2021) and QT (Hu et al. 2024a) for offline PbRL. The more implementation details are shown in Appendix D.

It is worth emphasizing that we provide a structural comparison of reward model between MLP and Transformer for offline PbRL. We find that that MLP-based reward model has a more stable performance than the Transformer-based one. Please see Appendix E2 for the detailed analysis. Hence, our methods employ the MLP-based reward model.

Q1: Can DTR Mitigate the Risk of Reward Bias and Lead to Enhanced Performance?

Gym-MuJoCo Locomotion Tasks. In the MuJoCo environment, we comprehensively compare DTR with the above baselines. The results are listed in Table 1. DTR outperforms all prior methods in most datasets. On average, DTR surpasses the best baseline FTB by a large margin with a minimum of 7.1% and even 11.0% in our best performance. Additionally, DTR exhibits significantly lower performance variance compared to Pb-DT and Pb-QT, which suffer from pronounced fluctuations. The superior performance reflects the importance of integrating CSM and TDL to mitigate the potential risk of reward bias in offline PbRL.

Adroit Manipulation Platform. We further evaluate DTR on the more challenging Adroit manipulation platform. We mainly compare DTR with Pb-IQL since it performs best on this benchmark in previous work (Yuan et al. 2024). with an average score higher than IQL by 28.9%.



(a) Compare of Reward Normalization. (b) Ablation with Ground-Truth Reward. (c) Scaling Trend for Preference Dataset.

Figure 4: Visual comparison of ablation experiments. (a) Changes in the reward values of trajectories labeled with different reward normalization methods in Halfcheetah-m-e; (b) The impact of the proposed module on the performance under the GT rewards and estimated rewards datasets. We record the average score across all 9 tasks of MuJoCo and "Base" represents DTR without the dynamic coefficient and ensemble normalization; (c) The performance of different methods changes with the size of the preference dataset. We record the average score across 3 tasks of MuJoCo-Medium.

Q2: What Specific Roles Do the Proposed Modules Play in DTR?

The Role of Dynamic Coefficient. To further elucidate the role of the proposed dynamic coefficients $\lambda(\text{step})$ in Equation 6 and 7. We study the performance difference compared to $\eta = 0$ and $\eta = \text{const}$, and present the training score curves in Figure 3a. The results of first 15k training steps highlight the superior training efficiency of dynamic coefficient, while the method with constant η fluctuates and performs lower than the dynamic coefficient one.

Comparison of Reward Normalization. To explore the effectiveness of the proposed ensemble normalization of rewards, we present the training score curves for different normalization methods in Figure 3b. The curves demonstrate that the ensemble normalization method achieves superior convergence and reduced training fluctuations. Furthermore, we select the first 100 timesteps of the trajectory with the lowest GT return in Halfcheetah-m-e dataset and plot the changes in different estimated rewards in Figure 4a. Compared to GT rewards, the original rewards (MLP) have little differences across states, which is detrimental to downstream policy learning. Directly normalizing average rewards amplifies these differences but also exaggerates estimation errors, especially overestimation. Our method normalizes the estimates of each ensemble member individually before averaging them, thereby enhancing reward differences while mitigating estimation errors.

Overall Ablation Performance. Table 3 records the average scores of the above two ablation experiments, and more ablation results are given in Appendix E5.

Ablation with GT Reward. To explore the ability of DTR in the offline RL domain, we apply the same dynamic coefficient and normalization modules as DTR on the offline dataset with GT rewards and show the results in Figure 4b. With GT rewards, the method incorporating the dynamic coefficient η performs closely to the oracle method. However, the performance significantly drops with reward normalization, likely due to inaccurate value estimation caused by the normalization of precise rewards. DTR, which employs preference learning to estimate rewards, achieves performance comparable to the oracle.

Gym-MuJoCo	DTR	-dynamic η	-reward_norm
Average	82.88 ± 1.20	79.63 ± 2.27 (-3.9%)	80.30 ± 0.77 (-3.1%)

Table 3: The Gym-MoJoCo average score of DTR without dynamic coefficient ($\eta = \text{const}$) and reward normalization.

Q3: Can DTR Still Perform Well With Small Amounts of Preference Feedback?

Human preference queries are often limited, so effective algorithms should align with human intentions using fewer queries. Consequently, we conduct experiments to evaluate the scalability of DTR and baselines with varying preference dataset scales in the MuJoCo-medium environment. We respectively train the reward model with portions of the queries from a dataset of 2000 preference pairs and then train downstream offline RL. The evaluation results are reported in Figure 4c. It shows that DTR can also achieve good performance when reducing the amount of preference data. indicating that finer reward estimation aids in enhancing the performance of the TDL module.

Conclusion

In this work, we propose DTR, which dynamically combines DT and TDL and utilizes the in-dataset trajectory returns and ensemble reward normalization to alleviate the risk of reward overestimation in offline PbRL. We show the potential of CSM in offline PbRL, which may motivate further research on generative methods in this field and extend them to online settings. In the future, combining the bi-level optimization of the reward function and the policy will be an interesting topic.

Acknowledgements

This work is supported by the National Key Research and Development Program of China under Grants 2022YFA1004000, the National Natural Science Foundation of China under Grants 62173325, and the CAS for Grand Challenges under Grants 104GJHZ2022013GC.

References

An, G.; Lee, J.; Zuo, X.; Kosaka, N.; Kim, K.-M.; and Song, H. O. 2023. Direct preference-based policy optimization without reward modeling. *Advances in Neural Information Processing Systems*, 36: 70247–70266.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Brandfonbrener, D.; Bietti, A.; Buckman, J.; Laroche, R.; and Bruna, J. 2022. When does return-conditioned supervised learning work for offline reinforcement learning? *Advances in Neural Information Processing Systems*, 35: 1542–1553.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. *arXiv:1606.01540*.

Chebotar, Y.; Vuong, Q.; Hausman, K.; Xia, F.; Lu, Y.; Irpan, A.; Kumar, A.; Yu, T.; Herzog, A.; Pertsch, K.; et al. 2023. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning*, 3909–3928. PMLR.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Chen, Y.; Li, H.; and Zhao, D. 2024. Boosting Continuous Control with Consistency Policy. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 335–344.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Early, J.; Bewley, T.; Evers, C.; and Ramchurn, S. 2022. Non-markovian reward modelling from trajectory labels via interpretable multiple instance learning. *Advances in Neural Information Processing Systems*, 35: 27652–27663.

Emmons, S.; Eysenbach, B.; Kostrikov, I.; and Levine, S. 2022. RvS: What is Essential for Offline RL via Supervised Learning? In *International Conference on Learning Representations*.

Fang, X.; Zhang, Q.; Gao, Y.; and Zhao, D. 2022. Offline reinforcement learning for autonomous driving with real world driving data. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 3417–3422. IEEE.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

Fujimoto, S.; and Gu, S. S. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34: 20132–20145.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

Gao, C.-X.; Fang, S.; Xiao, C.; Yu, Y.; and Zhang, Z. 2024. Hindsight Preference Learning for Offline Preference-based Reinforcement Learning. *arXiv preprint arXiv:2407.04451*.

Hejna, J.; Rafailov, R.; Sikchi, H.; Finn, C.; Niekum, S.; Knox, W. B.; and Sadigh, D. 2024. Contrastive Preference Learning: Learning from Human Feedback without Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

Hu, H.; Yang, Y.; Zhao, Q.; and Zhang, C. 2023. The Provable Benefit of Unsupervised Data Sharing for Offline Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.

Hu, S.; Fan, Z.; Huang, C.; Shen, L.; Zhang, Y.; Wang, Y.; and Tao, D. 2024a. Q-value Regularized Transformer for Offline Reinforcement Learning. In *Forty-first International Conference on Machine Learning*.

Hu, X.; Li, J.; Zhan, X.; Jia, Q.-S.; and Zhang, Y.-Q. 2024b. Query-Policy Misalignment in Preference-Based Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

Jin, C.; Krishnamurthy, A.; Simchowitz, M.; and Yu, T. 2020. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, 4870–4879. PMLR.

Kim, C.; Park, J.; Shin, J.; Lee, H.; Abbeel, P.; and Lee, K. 2023. Preference Transformer: Modeling Human Preferences using Transformers for RL. In *The Eleventh International Conference on Learning Representations*.

Kim, J.; Lee, S.; Kim, W.; and Sung, Y. 2024. Decision ConvFormer: Local Filtering in MetaFormer is Sufficient for Decision Making. In *The Twelfth International Conference on Learning Representations*.

Kostrikov, I.; Nair, A.; and Levine, S. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191.

Kumar, V. 2016. *Manipulators and Manipulation in high dimensional spaces*. Ph.D. thesis, University of Washington, Seattle.

Lee, K.; Smith, L.; and Abbeel, P. 2021. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*.

Liang, X.; Shu, K.; Lee, K.; and Abbeel, P. 2022. Reward Uncertainty for Exploration in Preference-based Reinforcement Learning. In *International Conference on Learning Representations*.

Liu, M.; Zhu, Y.; Chen, Y.; and Zhao, D. 2024. Enhancing Reinforcement Learning via Transformer-based State Predictive Representations. *IEEE Transactions on Artificial Intelligence*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Shang, J.; Li, X.; Kahatapitiya, K.; Lee, Y.-C.; and Ryoo, M. S. 2022. Starformer: Transformer with state-action-reward representations for robot learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(11): 12862–12877.

Shin, D.; Dragan, A.; and Brown, D. S. 2023. Benchmarks and Algorithms for Offline Preference-Based Reward Learning. *Transactions on Machine Learning Research*.

Skalse, J.; Howe, N.; Krasheninnikov, D.; and Krueger, D. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35: 9460–9471.

Sutton, R. S. 2018. Reinforcement learning: An introduction. *A Bradford Book*.

Verma, M.; and Metcalf, K. 2024. Hindsight PRIORs for Reward Learning from Human Preferences. In *The Twelfth International Conference on Learning Representations*.

Wang, J.; Zhang, Q.; Mu, Y.; Li, D.; Zhao, D.; Zhuang, Y.; Luo, P.; Wang, B.; and Hao, J. 2024a. Prototypical Context-Aware Dynamics for Generalization in Visual Control With Model-Based Reinforcement Learning. *IEEE Transactions on Industrial Informatics*.

Wang, Y.; Yang, C.; Wen, Y.; Liu, Y.; and Qiao, Y. 2024b. Critic-guided decision transformer for offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15706–15714.

Xu, S.; Fu, W.; Gao, J.; Ye, W.; Liu, W.; Mei, Z.; Wang, G.; Yu, C.; and Wu, Y. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

Yamagata, T.; Khalil, A.; and Santos-Rodriguez, R. 2023. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*, 38989–39007. PMLR.

Yu, T.; Kumar, A.; Chebotar, Y.; Hausman, K.; Finn, C.; and Levine, S. 2022. How to leverage unlabeled data in offline reinforcement learning. In *International Conference on Machine Learning*, 25611–25635. PMLR.

Yuan, Y.; Jianye, H.; Ma, Y.; Dong, Z.; Liang, H.; Liu, J.; Feng, Z.; Zhao, K.; and ZHENG, Y. 2024. Uni-RLHF: Universal Platform and Benchmark Suite for Reinforcement Learning with Diverse Human Feedback. In *The Twelfth International Conference on Learning Representations*.

Zeng, Z.; Zhang, C.; Wang, S.; and Sun, C. 2024. Goal-conditioned predictive coding for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Zhan, W.; Uehara, M.; Kallus, N.; Lee, J. D.; and Sun, W. 2024. Provable Offline Preference-Based Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*.

Zhang, Z.; Sun, Y.; Ye, J.; Liu, T.-S.; Zhang, J.; and Yu, Y. 2023. Flow to better: Offline preference-based reinforcement learning via preferred trajectory generation. In *The Twelfth International Conference on Learning Representations*.

Zhu, B.; Jordan, M.; and Jiao, J. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning*, 43037–43067. PMLR.