Reward Model Evaluation via Automatically-Ranked Policy Alignment

Aoran Wang, Lei Ou, Yang Yu, Zongzhang Zhang*

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
School of Artificial Intelligence, Nanjing University, Nanjing 210023, China
{wangar, oul}@lamda.nju.edu.cn, {yuy, zhangzz}@nju.edu.cn

Abstract

Evaluating reward models is a fundamental challenge in Reinforcement Learning (RL), particularly in settings where the reward model is learned or manually designed. The standard paradigm for Reward Model Evaluation (RME) involves training an optimal policy via RL on the given reward model and assessing model quality through the performance of the resulting policy. However, this approach conflates the quality of the reward model with the effectiveness of RL training, and is computationally expensive due to the need for policy optimization. Recent RME methods attempt to circumvent this issue by evaluating reward models directly, without RL, but often rely on impractical assumptions such as access to a ground-truth reward or fail to utilize available supervision in a fine-grained manner. To overcome these limitations, we propose the Policy Preference Alignment Coefficient (PPAC), a novel metric for RME that requires neither RL training nor ground-truth rewards. PPAC first generates a sequence of automatically ranked policy preferences that guarantee monotonic improvement in the policy value, and then quantifies the alignment between these generated preferences and those implied by the candidate reward model. Experimental results across gridworld and continuous control task demonstrate that PPAC yields preference sequences with consistently increasing policy values and outperforms existing metrics in evaluating reward model quality.

1 Introduction

In Reinforcement Learning (RL), the Reward Model (RM) plays a central role in shaping the behavior of the learned policy (Sutton and Barto 1998). In many real-world applications—such as autonomous driving (Codevilla et al. 2018) and language model alignment (Ouyang et al. 2022)—the ground-truth reward is unavailable, making it necessary to construct a surrogate reward model. This can be achieved through expert-crafted heuristics, language model-assisted design (Chen et al. 2024), or data-driven methods such as inverse reinforcement learning (Ng and Russell 2000) and preference-based learning (Christiano et al. 2017; Bai et al. 2022). However, the absence of a ground-truth reward gives rise to a fundamental challenge: how can we evaluate the quality of a learned or designed reward model? This question highlights the need for effective and practical Reward Model Evaluation (RME) techniques.

A prevalent RME paradigm assesses a reward model by training an optimal policy via RL under the given reward and measuring the performance of the resulting policy through metrics such as cumulative return or task success rate (Booth et al. 2023). Under this paradigm, reward models are deemed superior if they induce higher-performing policies. However, this evaluation strategy conflates reward model quality with the effectiveness of RL training. For example, a high-quality reward model might appear poor due to unstable RL optimization—caused by algorithmic brittleness, suboptimal hyperparameters, or stochasticity in training (Engstrom et al. 2019)—while a flawed reward model might appear strong due to reward hacking (Skalse et al. 2022), where the agent exploits reward artifacts to achieve high returns. Moreover, this approach is computationally expensive, as it requires retraining policies for each candidate RM. This cost becomes prohibitive in large-scale settings, such as Reinforcement Learning from Human Feedback (RLHF) for large language models (Ouyang et al. 2022; Shao et al. 2024), or when evaluating a large ensemble of reward models.

To mitigate these challenges, recent RME works explore RL-free methods that disentangle RME from policy optimization. One line of research proposes discrepancy-based metrics that directly compare the canonicalized candidate reward model against a known ground-truth reward (Gleave et al. 2021; Wulfe et al. 2022; Skalse et al. 2024). These methods often employ shaping-invariant metrics, offering theoretically sound comparisons. However, their reliance on the access to the ground-truth reward—what is unavailable in most practical scenarios—limits their applicability.

Another class of methods evaluates reward models directly against available supervision signals, such as expert demonstrations or trajectory preferences (Brown et al. 2021; Muslimani et al. 2025). These approaches are more practical, as such supervision is often accessible in real-world settings. For instance, the Trajectory Alignment Consistency (TAC) (Muslimani et al. 2025) assesses a reward model by comparing the pairwise preferences it induces with those present in the data. While TAC avoids the high cost of RL training, its reliance on pairwise comparison alone fails to

*Zongzhang Zhang is the corresponding author.

capture finer distinctions in reward quality (Zhu, Jordan, and Jiao 2023; Choi et al. 2024). In particular, methods limited to binary preferences may struggle to differentiate high-quality reward models from moderately good ones.

To address the limitations of existing RME approaches, we propose the Policy Preference Alignment Coefficient (PPAC), a novel RME metric that enables fine-grained evaluation of reward models using only trajectory comparisons. Unlike prior methods, PPAC does not rely on RL training or access to ground-truth rewards, making it practical for real-world applications with RME demand such as preference-based RL and human feedback alignment.

Contribution At the core of PPAC is our proposed policy improvement criterion Demonstration-Guided Policy Improvement (DGPI). DGPI iteratively generates a sequence of policies with monotonically increasing value under the unknown ground-truth reward by leveraging expert-preferred trajectories. This is achieved by estimating a surrogate advantage function based on the expert's state-value function, allowing each policy update to move toward expert-like behavior without explicitly recovering the reward. The resulting policy sequence forms an automatically-ranked ordering that reflects increasing alignment with expert preferences.

Using this sequence, PPAC evaluates a candidate reward model by comparing its induced policy preference ranking to the DGPI-generated ranking. Specifically, PPAC computes the Spearman's rank correlation coefficient between the two rankings, capturing the alignment between the candidate reward model and the expert-guided preferences.

We validate PPAC on grid-world navigation and continuous control task. Experimental results show that PPAC more accurately reflects the underlying quality of candidate reward models compared to existing baselines, providing an efficient tool for direct RME from trajectory comparisons.

2 Preliminaries

Notations We consider the Markov Decision Process (MDP) (Sutton and Barto 1998) formalized as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, d_0, r, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, d_0 is the initial state distribution, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1]$ is the discount factor.

Given a stationary stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, the discounted state distribution is defined as: $d_\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\pi(s_t = s)$, which reflects the discounted visitation frequency of state s under policy π. The occupancy measure of π is defined as: $\rho_\pi(s, a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\pi(s_t = s, a_t = a) = d_\pi(s) \cdot \pi(a \mid s)$, which gives the joint discounted visitation frequency of state-action pairs. For simplicity, we denote expectations over this distribution as $\mathbb{E}_\pi[\cdot] \coloneqq \mathbb{E}_{(s,a)\sim\rho_\pi}[\cdot]$ throughout the paper.

The expected return of policy π under reward function r is defined as the expected discounted sum of rewards:

$$J(r, \pi) = \mathbb{E}_\pi \left[r\left(s, a\right) \right].$$

The RL objective is to find an optimal policy π^* in the policy space Π that maximizes the expected return $J(r, \pi)$:

$$\pi^* = \arg\max_{\pi \in \Pi} J(r, \pi).$$

The state-action value function Q^π and the state value function V^π of a policy π are defined as

$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \left[V^\pi\left(s'\right) \right],$$
$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[Q^\pi(s, a) \right].$$

The advantage function of π is given by:

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s).$$

Problem Statement We assume access to a preference dataset $\mathcal{D} = \{(\tau_i^+, \tau_i^-)\}_{i=1}^N$ consisting of N trajectory comparisons. In each comparison pair, τ_i^+ denotes the preferred (or better) trajectory and τ_i^- the rejected (or worse) trajectory. The two trajectories in each comparison pair share the same initial state. We make two key assumptions:

- All preferred trajectories $\tau_i^+ = (s_0^i, a_0^i, s_1^i, \ldots, s_T^i)$ are generated by a common stationary stochastic policy π^+ in a latent MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, d_0, r_{\text{gt}}, \gamma \rangle$, and all rejected τ_i^- are generated similarly by another policy π^-.
- The policy π^+ corresponds to an expert policy π_E that is optimal under the unknown ground-truth reward r_{gt}.

These assumptions are plausible in practice, as $\{\tau_i^+\}_{i=1}^N$ typically represents the full set of demonstrations of desirable behavior, and in the absence of further supervision, we treat them as approximating the expert behavior. Conversely, $\{\tau_i^-\}_{i=1}^N$ represents the suboptimal alternatives.

The goal of this paper is to define a metric $M(r_{\text{eval}}, \mathcal{D})$ that evaluates the quality of a candidate reward model r_{eval} using only the preference dataset $\mathcal{D}$. As motivated in Section 1, we aim for the metric that avoids RL training on r_{eval}, and exploits the implicit supervision embedded in trajectory preferences, beyond merely classifying τ_i^+ vs. τ_i^-.

3 Method

We introduce our proposed RME framework based on the automatically-ranked policy preferences derived from trajectory comparisons. This section is structured as follows:

- We define policy preferences based on expected state value under initial state distribution, extending prior formulations (Bowling et al. 2023; Muslimani et al. 2025).
- We propose Demonstration-Guided Policy Improvement (DGPI), a policy update criterion that guarantees monotonic improvements in policy value.
- We describe the process of generating automatically-ranked policy sequences with DGPI and computing the Policy Preference Alignment Coefficient (PPAC).

An overview of the complete PPAC computation pipeline is illustrated in Figure 1. Detailed derivations and proofs in this section are deferred to Appendix A [1].

Policy Preference Let the discounted state visitation probability from an initial state s be defined as: $\mathbb{P}_t(s') = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\pi(s_t = s' | s_0 = s)$. Then considering following the policy π from s, the state value function $V^\pi(s)$ under the ground-truth reward r_{gt} is given by:

$$V^\pi(s) = \mathbb{E}_{s_t \sim \mathbb{P}_t, a_t \sim \pi(\cdot|s_t)} \left[r_{\text{gt}}\left(s_t, a_t\right) \right]. \quad (1)$$

[1] https://www.lamda.nju.edu.cn/wangar/PPAC_supp.pdf

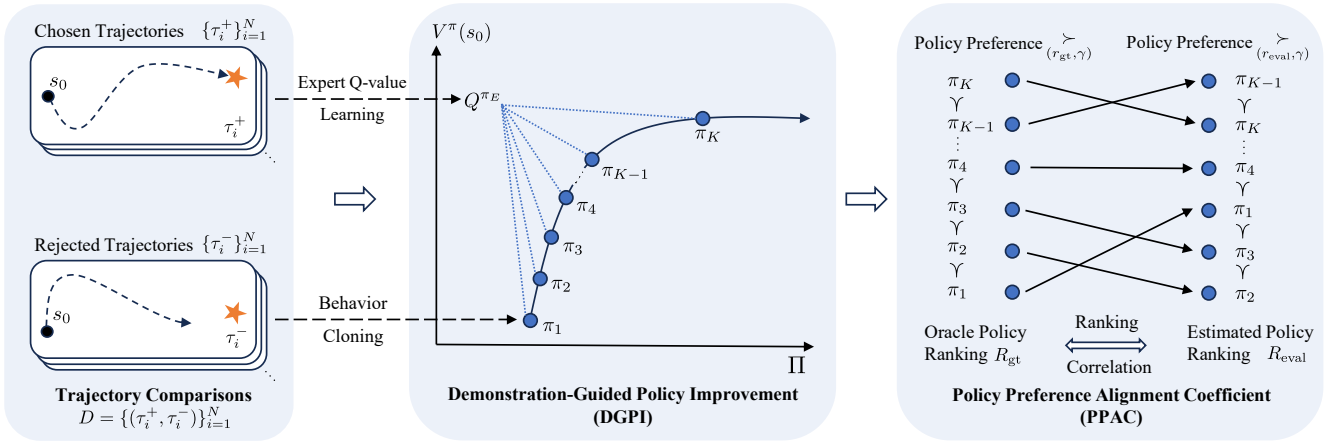

Figure 1: Overview of PPAC computation from trajectory comparison supervision. DGPI generates an automatically-ranked sequence of policies using the expert Q-value learned from the preferred trajectories. PPAC evaluates the quality of a candidate reward model by measuring the alignment between its induced policy rankings and the oracle ranking produced by DGPI.

We further define the support set of initial state distribution $\mathcal{S}_{\text{init}} \subseteq \mathcal{S}$ as the following set of states,

$$\mathcal{S}_{\text{init}} = \{s \in \mathcal{S} \mid d_0(s) > 0\}. \tag{2}$$

We can now define the preference between policies, i.e., the policy preference, based on Eq. 1 and Eq. 2.

Definition 3.1 (Policy Preference)**.** *In an infinite-horizon MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, d_0, r_{\text{gt}}, \gamma \rangle$, we define the preference ordering over stationary stochastic policies as follows:*

$$\pi_A \underset{(r_{\text{gt}},\gamma)}{\succ} \pi_B \iff V^{\pi_A}(s_0) > V^{\pi_B}(s_0), \text{ for all } s_0 \in \mathcal{S}_{init}.$$

That is, policy π_A is preferred over policy π_B if and only if starting from every possible initial state, the expected return of the trajectories following π_A is larger than the expected return of the trajectories following π_B.

The subscript (r_{gt}, γ) emphasizes that this policy preference ordering is with respect to the reward r_{gt} and the discount factor γ. In the rest of the paper, we simplify this policy preference w.r.t. (r_{gt}, γ) as $\succ$ when there is no ambiguity. Since π^+ is assumed to be optimal under the reward r_{gt}, it follows directly that π^+ is preferred over π^-, i.e., $\pi^+ \succ \pi^-$.

Demonstration-Guided Policy Improvement (DGPI) We now introduce DGPI, a novel policy improvement rule guided by expert demonstrations that ensures monotonic improvements in the policy value, even in the absence of r_{gt}.

We begin with a classic result from (Kakade and Langford 2002) that measures the expected value difference.

Lemma 3.2. *Given an initial state distribution d_0, for any two policies $\tilde{\pi}$ and π,*

$$\mathbb{E}_{s\sim d_0}\left[V^{\tilde{\pi}}(s)\right] - \mathbb{E}_{s\sim d_0}\left[V^{\pi}(s)\right] = \frac{1}{1-\gamma}\mathbb{E}_{\tilde{\pi}}\left[A^{\pi}(s,a)\right]. \tag{3}$$

To improve from π within the trust-region constraint (Schulman et al. 2015), we consider the following optimization objective that maximizes the right-hand side of Eq. 3 with a KL-divergence regularization:

$$\arg\max_{\tilde{\pi}\in\Pi} \mathbb{E}_{\tilde{\pi}}\left[A^{\pi}(s,a)\right] - \alpha D_{\text{KL}}^{d_{\tilde{\pi}}}(\tilde{\pi} \,\|\, \pi), \tag{4}$$

where $D_{\text{KL}}^{d_{\tilde{\pi}}}(\tilde{\pi}\|\pi) = \mathbb{E}_{s\sim d_{\tilde{\pi}}}[D_{\text{KL}}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s))]$ is the expected KL-divergence between $\tilde{\pi}(\cdot|s)$ and $\pi(\cdot|s)$ under the state distribution $d_{\tilde{\pi}}$. The maximization in Eq. 4 admits an analytical solution (Azar, Gómez, and Kappen 2012):

$$\tilde{\pi}(a \mid s) = \frac{1}{Z(s)}\pi(a \mid s)\exp\left(\frac{1}{\alpha}A^{\pi}(s,a)\right), \tag{5}$$

where the state-dependent function $Z(s)$ is the normalizer to ensure that $\tilde{\pi}(a|s)$ sums up to 1 on all actions for each s.

Proposition 1 in (Wang et al. 2018) indicates that the updated policy $\tilde{\pi}$ in Eq. 5 ensures: for any state $s \in \mathcal{S}$, $V^{\tilde{\pi}}(s) > V^{\pi}(s)$ before convergence. In other words, updating π with Eq. 5 naturally yields a policy preference $\tilde{\pi} \succ \pi$.

However, we can not thus construct automatically-ranked policy preferences $\pi^- = \pi_1 \prec \pi_2 \prec \cdots \prec \pi_K = \pi^+$ by iteratively updating with Eq. 5. The main challenge is that without knowing r_{gt}, we can not estimate the advantage function $A^{\pi}(s,a)$. In fact, recovering the exact ground-truth reward from demonstration supervision remains an open and ill-posed problem (Lazzati and Metelli 2025).

Despite the inaccessibility of exact r_{gt}, recent work has shown that it is possible to estimate the optimal state-action value function $Q^{\pi_E}(s,a)$ corresponding to the expert policy π^E, purely from the expert demonstrations (Garg et al. 2021; Sikchi et al. 2024; Moulin, Neu, and Viano 2025). This observation opens a promising pathway for policy improvement without the ground-truth reward recovery. One representative approach is Inverse soft-Q Learning (IQ-Learn) (Garg et al. 2021), which formulates a non-adversarial method to recover $Q^{\pi_E}(s,a)$. However, IQ-Learn inherently depends on entropy regularization and assumes the policy is implicitly shaped by a regularized reward, which is not aligned with our unregularized setup. In contrast, Saddle-Point Offline Imitation Learning (SPOIL) learns $Q^{\pi_E}(s,a)$

Algorithm 1: SPOIL (Moulin, Neu, and Viano 2025)

Input: Expert trajectories τ_E, learning rate $\eta > 0$, and numbers of iteration T

Initialize Q-function Q_0, uniform policy π_0.
for step $t = 1, 2, \ldots, T$ **do**
  $\pi_t(a \mid s) \propto \pi_{t-1}(a \mid s) \exp(\eta \cdot Q_{t-1}(s, a))$.
  Train $Q_t(s, a)$ by solving the following maximization $\arg\max_Q \mathbb{E}_{\pi_E}\big[Q(s, a) - \mathbb{E}_{a' \sim \pi_t}[Q(s, a')]\big]$.
end for
return Q_T

by alternating between Q-value estimation and policy optimization in the following formulation without any regularization on policy or reward model:

$$Q_k(s, a) \in \arg\max_Q \mathbb{E}_{\pi_E}\big[Q(s, a) - \mathbb{E}_{a' \sim \pi_k}[Q(s, a')]\big],$$

$$\pi_{k+1}(a \mid s) = \frac{\pi_k(a \mid s)\, \exp\big(\eta \cdot Q_k(s, a)\big)}{\sum_{a' \in \mathcal{A}} \pi_k(a' \mid s)\, \exp\big(\eta \cdot Q_k(s, a')\big)},$$

where $\eta > 0$ is the learning rate controlling the update strength. As shown in prior works (Ho and Ermon 2016; Garg et al. 2021), this iterative scheme converges to the optimal state-action value function $Q^* = Q^{\pi_E}(s, a)$.

In our framework, SPOIL is used as a subroutine to recover $Q^{\pi_E}(s, a)$ from the expert trajectories $\{\tau_i^+\}_{i=1}^N$. The SPOIL algorithm is summarized in Algorithm 1, and more details on it are deferred to Appendix B.

Being able to obtain $Q^{\pi_E}(s, a)$, we define the surrogate value function $\hat{V}^\pi(s)$ and surrogate advantage function $\hat{A}^\pi(s, a)$ with respect to the policy π as follows:

$$\begin{aligned} \hat{V}^\pi(s) &= \mathbb{E}_{a \sim \pi(\cdot|s)}\left[Q^{\pi_E}(s, a)\right], \\ \hat{A}^\pi(s, a) &= Q^{\pi_E}(s, a) - \hat{V}^\pi(s). \end{aligned} \tag{6}$$

Intuitively, $\hat{V}^\pi(s)$ represents the expected return of executing policy π for one step from the state s, then following π^E thereafter. The surrogate advantage $\hat{A}^\pi(s, a)$ reflects the benefit of consistently following π^E from the state s instead of taking an action $a \sim \pi(\cdot|s)$ at the first step.

In analogy to Lemma 3.2, we establish that the surrogate advantage estimates above yield valid improvement signals:

Lemma 3.3. *For any two policies $\tilde{\pi}$ and π,*

$$\mathbb{E}_{s \sim d_{\tilde{\pi}}}\left[\hat{V}^{\tilde{\pi}}(s)\right] - \mathbb{E}_{s \sim d_{\tilde{\pi}}}\left[\hat{V}^{\pi}(s)\right] = \frac{1}{1-\gamma}\mathbb{E}_{\tilde{\pi}}\left[\hat{A}^{\pi}(s, a)\right]. \tag{7}$$

We use this result to derive a surrogate policy update rule analogous to Eq. (5), replacing the true advantage with its surrogate counterpart. To obtain a policy $\tilde{\pi}$ that maximally improves over π, we maximize the right-hand side of Eq. 8 with KL-regularization $D_{\mathrm{KL}}^{d_{\tilde{\pi}}}(\tilde{\pi} \,\|\, \pi)$, yielding the following closed-form solution:

$$\tilde{\pi}(a \mid s) = \frac{1}{Z'(s)}\pi(a \mid s) \exp\big(\frac{1}{\alpha}\hat{A}^{\pi}(s, a)\big), \tag{8}$$

where $\hat{A}^\pi(s, a)$ is the surrogate advantage function in Eq. 6, and $Z'(s)$ is the normalizer that ensures $\tilde{\pi}(a|s)$ forms a valid probability distribution. Our proposed Demonstration-Guided Policy Improvement (DGPI) adopts this surrogate-advantage-weighted updating as its core mechanism.

The following proposition confirms that DGPI enjoys the same monotonic improvement guarantee as the update in Eq. 5, despite using surrogate advantage estimates:

Proposition 3.4 (Monotonic Improvement)**.** *Let $\tilde{\pi}$ be obtained from π via Eq. 8. Then:*

$$V^{\tilde{\pi}}(s) > V^{\pi}(s), \quad \textit{for all } s \in \mathcal{S}.$$

That is, $\tilde{\pi} \succ \pi$ w.r.t. the r_{gt} and the discounted factor γ.

Moreover, repeated application of exact DGPI from any initial policy π_1 yields a sequence of strictly improving policies $\pi_1, \pi_2, \ldots$, eventually converging to expert policy π_E:

Proposition 3.5. *Let $\{\pi_t\}_{t \geq 1}$ denote the sequence produced by iteratively applying exact DGPI. Then for some T, we have $\pi_T = \pi_E$, and $\pi_1 \prec \pi_2 \prec \cdots \prec \pi_T$.*

These properties of DGPI enable us to automatically generate a preference chain of policies $\pi^- = \pi_1 \prec \pi_2 \prec \cdots \prec \pi_K = \pi^+$ purely from trajectory preference data, laying the foundation for our proposed PPAC metric.

Policy Preference Alignment Coefficient (PPAC) We now describe how to use DGPI to generate a sequence of automatically-ranked policy preferences from the preference dataset $\mathcal{D} = \{(\tau_i^+, \tau_i^-)\}_{i=1}^N$, and how to compute our proposed evaluation metric—Policy Preference Alignment Coefficient (PPAC)—based on the alignment between this policy sequence and a candidate reward model.

Motivated by recent findings that listwise comparisons provide richer supervision than pairwise preferences for reward learning (Zhu, Jordan, and Jiao 2023; Choi et al. 2024), we hypothesize that enforcing consistency with a listwise ordering over policies enables finer-grained evaluation of reward model quality, compared with a pairwise ordering.

To construct such a listwise ordering, we begin by initializing the policy sequence from the rejected policy $\pi_1 = \pi^-$, which we approximate using Behavior Cloning (BC) (Pomerleau 1991) on the set of rejected trajectories $\{\tau_i^-\}_{i=1}^N$ Then, we apply DGPI iteratively to obtain the subsequent policies $\pi_1, \pi_2, \ldots, \pi_K$, where each update is given by:

$$\tilde{\pi}(a \mid s) = \frac{1}{Z'(s)}\pi_k(a \mid s) \exp\big(\frac{1}{\alpha}\hat{A}^{\pi_k}(s, a)\big). \tag{9}$$

In practice, we fit the parameterized π_{k+1} by minimizing the KL-divergence between the DGPI target $\tilde{\pi}$ and π_{k+1}:

$$\begin{aligned} &\arg\min_{\pi_{k+1} \in \Pi} D_{\mathrm{KL}}^{d_{\pi_k}}(\tilde{\pi} \| \pi_{k+1}) \\ &= \arg\max_{\pi_{k+1} \in \Pi} \sum_s d_{\pi_k}(s) \sum_a \tilde{\pi}(a \mid s) \log \pi_{k+1}(a \mid s) \\ &\approx \arg\max_{\pi_{k+1} \in \Pi} \mathbb{E}_{\pi_k}\left[\exp\left(\frac{1}{\alpha}\hat{A}^{\pi_k}(s, a)\right) \log \pi_{k+1}(a \mid s)\right], \end{aligned} \tag{10}$$

which is equivalent to a weighted behavior cloning objective where samples are reweighted by $\exp\left(\frac{1}{\alpha}\hat{A}^{\pi_k}(s, a)\right)$.

We repeat this DGPI process in Eq. 10 to convergence. According to Proposition 3.5, we obtain a strictly ordered sequence of policy preferences:

$$\pi_1 = \pi^- \prec \pi_2 \prec \cdots \prec \pi_K = \pi^+,$$

which—by construction—satisfies a monotonic increase in their values under the ground-truth reward r_{gt} and starting from any possible initial state s in $\mathcal{D}$:

$$V^{\pi_1}(s) < \cdots < V^{\pi_K}(s).$$

This ranking of policy values provides a supervision signal at the level of policy preference, which we refer to as the oracle policy ranking R_{gt}. To evaluate a candidate reward model r_{eval}, we assess whether its induced preferences over the policies $\{\pi_1, \ldots, \pi_K\}$ aligns with the oracle ranking.

For each comparison pair $(\tau_i^+, \tau_i^-) \in \mathcal{D}$, we first identify the shared initial state s_i, and sample M trajectories from each policy π_k starting at s_i. The empirical value $V^{\pi_k}(s_i)$ of each policy under r_{eval} is estimated as:

$$V^{\pi_k}(s_i) = \frac{1}{M} \sum_{m=1}^{M} \sum_{t=0}^{T_m - 1} \gamma^t r_{\text{eval}}(s_t^m, a_t^m), \qquad (11)$$

where (s_t^m, a_t^m) denotes the t-th transition in the m-th trajectory sampled by π_k with horizon T_m. We denote the ranking over these estimated values as R^i_{eval}. If r_{eval} functions as good as r_{gt}, the rankings R^i_{eval} and R_{gt} should exactly match. On the contrary, the worse r_{eval} is than r_{gt}, the more severely the rankings R^i_{eval} and R_{gt} should differ. To quantify their alignment, we use Spearman's rank correlation coefficient:

$$C(R^i_{\text{eval}}, R_{\text{gt}}) = \frac{\text{cov}(R^i_{\text{eval}}, R_{\text{gt}})}{\sigma_{R^i_{\text{eval}}} \cdot \sigma_{R_{\text{gt}}}}, \qquad (12)$$

where $\text{cov}(R^i_{\text{eval}}, R_{\text{gt}})$ denotes the covariance of these two rank variables, and $\sigma_{R^i_{\text{eval}}}$, $\sigma_{R_{\text{gt}}}$ are the standard deviations.

The Policy Preference Alignment Coefficient (PPAC) is then defined as the average Spearman's rank correlation across all N comparison pairs in $\mathcal{D}$, i.e.,

$$\text{PPAC}(r_{\text{eval}}, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} C(R^i_{\text{eval}}, R_{\text{gt}}).$$

In practice, we monitor the surrogate state value $\hat{V}^{\pi_k}$ to select a set of K policies that are approximately evenly spaced in the value range, ensuring diverse preference supervision throughout the DGPI process. We show the process of computing PPAC from $\mathcal{D} = \{\tau_i^+, \tau_i^-\}_{i=1}^N$ in Algorithm 2.

By construction, A PPAC value near 1 indicates a strong agreement between the estimated and oracle rankings, reflecting high reward model fidelity. A PPAC value near -1 indicates a reverse alignment. Moreover, PPAC is designed to be invariant to common reward transformations, ensuring that it evaluates only the policy preference structure encoded by the reward model, rather than its numeric value.

Proposition 3.6. *When evaluated under the same preference dataset $\mathcal{D}$, if the reward models r and r' differ by a positive scaling transformation*

$$r'(s,a) = \beta_1 \cdot r(s,a) + \beta_2,$$

Algorithm 2: Computing the PPAC

Input: Candidate reward model r_{eval}, trajectory comparisons dataset $\mathcal{D} = \{\tau_i^+, \tau_i^-\}_{i=1}^N$, and granularity control parameter K

```
Estimate Q^{π_E} from {τ_i^+}_{i=1}^N with SPOIL. (Alg. 1)
Train policy π_1 from {τ_i^-}_{i=1}^N with BC.
Generate ranked policies π_1 ≺ ··· ≺ π_K in DGPI updating from π_1. (Eq. 10)
for i = 1 to N do
    Let s_i ← initial state of (τ_i^+, τ_i^-).
    for k = 1 to K do
        Sample trajectories starting from π_k starting at s_i.
        Estimate V^{π_k}(s_i) under r_eval. (Eq. 11)
    end for
    Compute C(R^i_eval, R_gt) between the estimated ranking R^i_eval and the oracle ranking R^i_gt. (Eq. 12)
end for
return PPAC(r_eval, D) = (1/N) Σ_{i=1}^N C(R^i_eval, R_gt).
```

where $\beta_1 > 0, \beta_2$ are constants, or a potential-based reward shaping transformation (Ng, Harada, and Russell 1999)

$$r'(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Phi(s') - \Phi(s),$$

where $\Phi : \mathcal{S} \to \mathbb{R}$ is a potential function, then

$$\text{PPAC}(r, \mathcal{D}) = \text{PPAC}(r', \mathcal{D}).$$

4 Experiment

In this section, we evaluate PPAC against several baseline methods for RME. We begin with a summary of the experimental setup, followed by main results. Additional implementation details and analysis are provided in Appendix C.

Benchmarks To assess whether PPAC serves as an effective RME metric, we evaluate it on two standard tasks:

- MiniGrid-DoorKey-8×8 (Chevalier-Boisvert et al. 2023): a goal-oriented navigation task in a 2D grid world with image-based observations and a discrete action space. The agent must first pick up a key and then open a door to reach the goal. The ground-truth reward is defined as $1 - 0.9 * (\text{step\_count}/\text{max\_steps})$ if the agent successfully reaches the goal, and 0 otherwise. We randomize 10 different instances of this task for experiment.
- HalfCheetah (MuJoCo) (Todorov, Erez, and Tassa 2012): a control task with feature-based observations and a continuous action space. The agent is rewarded for accelerating forward, with a penalty for excessive control inputs.

Baselines We compare PPAC with the following baselines:

- STAndardised Reward Comparison (STARC) (Skalse et al. 2024): STARC measures the discrepancy between a canonicalized candidate reward model and the ground-truth reward. It generalizes previous discrepancy-based metrics (Gleave et al. 2021; Wulfe et al. 2022), so we use its best-performing specification as the representative.
- TAC (Muslimani et al. 2025): TAC compares the pairwise preferences over trajectory distributions induced by the candidate reward and the supervision data.

Reward Model	NegativeSTARC	TAC	PPAC-BC	PPAC-AIL	PPAC
GroundTruth	0.0000	1.0000	0.8300	0.9400	0.9800
PotentialShaped	-1.3232	1.0000	0.9100	0.9100	0.9800
SecondGoal-Slight	-1.6591	1.0000	0.9100	0.8300	0.6400
SecondGoal-Intense	-1.7303	1.0000	0.8100	0.6220	0.4800
Constant	-1.6024	-1.0000	-0.9100	-0.9400	-0.9400

Table 1: RME results on the MiniGrid-DoorKey-8×8 task. Each value represents the average RME score across 10 randomized task instances. Higher values indicate candidate reward models of higher quality.

Reward Model	NegativeSTARC	TAC	PPAC-BC	PPAC-AIL	PPAC
GroundTruth	0.0000	1.0000	0.9353	0.9774	0.9873
PotentialShaped	-0.6745	1.0000	0.9353	0.9774	0.9873
SecondGoal-Slight	-0.2489	1.0000	0.8346	0.9774	0.7549
SecondGoal-Intense	-0.8099	1.0000	0.6060	0.9549	0.5203
Random	-1.8873	-1.0000	-0.9489	-0.9489	-0.9293

Table 2: RME results on the HalfCheetah task. Higher values indicate candidate reward models of higher quality.

- PPAC with noisy-injected Behavior Cloning (PPAC-BC): A variant of PPAC that replaces DGPI with noisy perturbations of the BC policy, inspired by Disturbance-based Reward Extrapolation (D-REX) (Brown, Goo, and Niekum 2020). Intermediate policies are generated by injecting increasing Gaussian noise into a BC policy trained on the chosen trajectories.
- PPAC with Adversarial Imitation Learning (PPAC-AIL): A variant of PPAC inspired by Automated Preference generation with Enhanced Coverage (APEC) (Zhang et al. 2025), whose intermediate policies are checkpoints in Adversarial Imitation Learning (AIL) training with the chosen trajectories serving as expert demonstrations.

For STARC, the ground-truth reward is exposed during evaluation as required. For PPAC-BC and PPAC-AIL, only the chosen trajectories are used as expert demonstrations. TAC and PPAC make full use of the comparison dataset.

All metrics except STARC yield scores between $[-1, 1]$, with higher values indicating better reward models. Since STARC is unbounded and higher values indicate worse performance, we report the negative STARC (i.e., NegativeSTARC) to maintain consistency in interpretation.

Experimental Design For each benchmark, we construct five reward models of descending quality:

- GroundTruth: the ground-truth RM, the best RM of all.
- PotentialShaped: the ground-truth RM transformed by potential-based shaping. This transformation preserves the optimal policy and should yield close RME score.
- SecondGoal-Slight: the ground-truth RM with an additional small bonus for achieving a secondary and orthogonal objective. In MiniGrid, this corresponds to reaching a distractor grid cell; in HalfCheetah, it incentivizes the agent maintaining a specific height.
- SecondGoal-Intense: the same as above but with a large bonus for the secondary goal, substantially diverting the agent's behavior in both MiniGrid and HalfCheetah.
- Constant (or Random): an uninformative reward, either randomly generated (or constant) across all transitions.

These five reward models above represent a quality spectrum from ideal to poor. An effective RME metric should assign similar scores to GroundTruth and PotentialShaped, and progressively lower scores to SecondGoal-Slight, SecondGoal-Intense, and Random at clear intervals.

Results We conduct experiments on 10 randomly generated DoorKey tasks. For each DoorKey task, we train a Proximal Policy Optimization agent (Schulman et al. 2017) under its ground-truth reward until it consistently reaches the goal, designating it as the latent chosen policy. A medium-performance checkpoint from the same training run—prior to convergence—is selected as the rejected policy. We collect $N = 5$ trajectories from each of these policies per task as supervision to compute TAC, PPAC-BC, PPAC-AIL, and PPAC. For PPAC-based methods, we set the ranking granularity to $K = 5$. For the HalfCheetah task, we train a Soft Actor-Critic agent (Haarnoja et al. 2018) that achieves a performance of approximately 12,000 as the chosen policy, and similarly select an intermediate policy of medium performance as the rejected one. We collect $N = 50$ trajectories from each policy and use a higher granularity with $K = 20$.

The RME results for DoorKey and HalfCheetah are reported in Table 1 and Table 2, respectively. We observe several key findings: NegativeSTARC, due to its unbounded scale, lacks interpretability to reward quality as an absolute metric. More critically, it fails to correctly recognize that PotentialShaped is as good as the GroundTruth, contradicting its theoretical invariance intention. TAC demonstrates limited discriminative power, failing to distinguish between all but the most degenerate reward model (Constant/Random). This supports our motivation that pairwise preference signals are insufficient for fine-grained RME. PPAC-BC performs reasonably well on HalfCheetah, correctly ordering most reward models. However, in DoorKey it incorrectly ranks PotentialShaped above both GroundTruth

and SecondGoal-Slight. Additionally, the low magnitude of PPAC-BC for GroundTruth suggests that the noise-injected policies used in this method do not consistently degrade in performance. This reflects the inherent instability in D-REX-style perturbations, where high-noise policies behave nearly randomly and fail to preserve performance ordering. PPAC-AIL successfully ranks reward models in DoorKey, but fails to do so in HalfCheetah by assigning similar scores to models of differing quality. This suggests that AIL-generated policy sequences lack sufficient value separation in high-dimensional continuous control settings. In contrast, PPAC consistently produces scores close to 1 for GroundTruth, correctly identifies PotentialShaped as equivalent to GroundTruth and clearly separates reward models across both tasks. These results indicate that PPAC generates more reliably ordered policy preferences, and provides more discriminative RME than existing baselines.

5 Related Work

Reward Model Evaluating Traditional RME relies on indirect evaluation, where the quality of a reward model is inferred from the performance of the optimal policy trained under it. In this paradigm, a reward model is considered superior if the policy trained via RL under that model achieves higher returns or task success rates. In contrast, PPAC provides a form of direct reward evaluation, decoupling reward model quality from the RL training dynamics. This separation is crucial, as it avoids confounding RME with RL instability, sensitivity to hyperparameters, or suboptimal policy convergence. Moreover, PPAC is highly efficient when multiple reward models must be evaluated, since DGPI is executed only once on the supervision dataset rather than requiring separate RL training for each reward candidate.

A number of prior works have proposed direct evaluation metrics that assess reward models without training policies. For example, Equivalent-Policy Invariant Comparison (EPIC) (Gleave et al. 2021) canonicalizes reward models by removing potential-based shaping and computes similarity using correlation over a fixed transition distribution. Dynamics-Aware Reward Distance (DARD) (Wulfe et al. 2022) extends EPIC by evaluating only on realizable transitions using known dynamics. STARC (Skalse et al. 2024) further generalizes EPIC and DARD into a unified framework with improved theoretical guarantees. However, all of these methods require access to the ground-truth reward, which is often unavailable in real-world tasks—the very reason why reward model learning is needed. In contrast, PPAC leverages trajectory comparisons, a form of supervision that is far more accessible in practice, especially in preference-based reward learning settings (Christiano et al. 2017).

Another line of work, such as TAC (Muslimani et al. 2025), also seeks to avoid RL by comparing rankings over trajectory distributions. TAC quantifies the alignment between the candidate reward model's induced preferences and an oracle ranking. While TAC provides a useful perspective, it operates on pairwise preferences, which often lack the granularity needed to discriminate between closely competing reward models. PPAC addresses this limitation by constructing automatically-ranked listwise policy sequences using our proposed DGPI procedure, thereby offering finer-grained RME. As shown in our experiments, preference misalignment is more easily captured through disorder in listwise preferences than through flips in pairwise comparisons.

Ranked Preference Generation A growing body of work demonstrates that listwise supervision offers significant advantages over pairwise comparisons in both reward learning and imitation learning (Zhu, Jordan, and Jiao 2023; Choi et al. 2024; Brown, Goo, and Niekum 2020; Zhang et al. 2025). These findings motivate the development of methods that generate automatically-ranked trajectories or policies from suboptimal data without requiring extra explicit labels or access to the ground-truth reward.

D-REX (Brown, Goo, and Niekum 2020) proposes generating ranked policies by injecting Gaussian noise of increasing magnitude into a BC policy trained on suboptimal demonstrations. While D-REX proves an upper bound on the expected return of such noise-injected policies, it does not guarantee monotonic degradation in performance as noise increases. APEC (Zhang et al. 2025), on the other hand, produces ranked policies by sampling intermediate policies from the training trajectory of AIL. Although APEC provides a lower bound on value during AIL training, it likewise lacks guarantees of monotonic improvement on the performance of generated policies. To the best of our knowledge, our proposed Demonstration-Guided Policy Improvement (DGPI) is the first method that ensures monotonic improvement in policy performance without requiring access to a ground-truth reward or external preference labeling. DGPI leverages only trajectory comparisons, making it well-suited to RME in preference-based reward learning.

While our primary focus is RME, DGPI is broadly applicable to other tasks that require generating ranked policies in the absence of the ground-truth reward. As shown in Appendix D, DGPI can be naturally extended to work under other types of supervision as well, making it a versatile tool in IL, curriculum generation, and structured exploration.

6 Conclusion and Limitation

We propose the PPAC, a novel RME metric without relying on RL training or access to ground-truth rewards. PPAC measures the alignment between a reward model's induced policy ranking and an expert-guided policy ranking generated via our proposed DGPI, which ensures monotonic policy improvement without extra supervision. Experiments on grid-world and continuous control tasks demonstrate that PPAC achieves more consistent and discriminative evaluation than existing RME methods, highlighting PPAC's capability as a practical, efficient, and theoretically grounded approach to reward model assessment.

One limitation of PPAC lies in its reliance on accurate estimation of the expert Q-function. When trajectory comparison data is limited, the unreliable estimation undermines DGPI's monotonic improvement guarantee. Additionally, PPAC is more of a necessity indicator of reward model quality as a high PPAC score certifies adequacy only within the coverage of the provided trajectories. In future studies, we will explore more on the coverage analysis to strengthen the sufficiency of our proposed RME indicator.

Acknowledgements

This work is supported by the National Science Foundation of China (62276126, 62250069, 62495093) and Jiangsu Science Foundation (BK20243039).

References

Azar, M. G.; Gómez, V.; and Kappen, H. J. 2012. Dynamic policy programming. *Journal of Machine Learning Research (JMLR)*, 13: 3207–3245.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Booth, S.; Knox, W. B.; Shah, J.; Niekum, S.; Stone, P.; and Allievi, A. 2023. The perils of trial-and-error reward design: misdesign through overfitting and invalid task specifications. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, 5920–5929.

Bowling, M.; Martin, J. D.; Abel, D.; and Dabney, W. 2023. Settling the reward hypothesis. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 3003–3020.

Brown, D. S.; Goo, W.; and Niekum, S. 2020. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Proceedings of the 2020 Conference on Robot Learning (CoRL)*, 330–359.

Brown, D. S.; Schneider, J.; Dragan, A.; and Niekum, S. 2021. Value alignment verification. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 1105–1115.

Chen, X.-H.; Wang, Z.; Du, Y.; Jiang, S.; Fang, M.; Yu, Y.; and Wang, J. 2024. Policy learning from tutorial books via understanding, rehearsing and introspecting. *Advances in Neural Information Processing Systems (NeuIPS)*, 37: 18940–18987.

Chevalier-Boisvert, M.; Dai, B.; Towers, M.; Perez-Vicente, R.; Willems, L.; Lahlou, S.; Pal, S.; and Castro, P. S. 2023. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems (NeuIPS)*, 36: 73383–73394.

Choi, H.; Jung, S.; Ahn, H.; and Moon, T. 2024. Listwise reward estimation for offline preference-based reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 8651–8671.

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 (NIPS)*, 4302–4310.

Codevilla, F.; Müller, M.; López, A.; Koltun, V.; and Dosovitskiy, A. 2018. End-to-end driving via conditional imitation learning. In *Proceedings of the 35th International Conference on Robotics and Automation (ICRA)*, 4693–4700.

Engstrom, L.; Ilyas, A.; Santurkar, S.; Tsipras, D.; Janoos, F.; Rudolph, L.; and Madry, A. 2019. Implementation matters in deep RL: A case study on PPO and TRPO. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

Garg, D.; Chakraborty, S.; Cundy, C.; Song, J.; and Ermon, S. 2021. IQ-Learn: Inverse soft-Q learning for imitation. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 4028–4039.

Gleave, A.; Dennis, M. D.; Legg, S.; Russell, S.; and Leike, J. 2021. Quantifying differences in reward functions. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 1861–1870.

Ho, J.; and Ermon, S. 2016. Generative adversarial imitation learning. *Advances in Neural Information Processing Systems (NIPS)*, 29: 4565–4573.

Kakade, S.; and Langford, J. 2002. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, 267–274.

Lazzati, F.; and Metelli, A. M. 2025. On the partial identifiability in reward learning: Choosing the best reward. *arXiv preprint arXiv:2501.06376*.

Moulin, A.; Neu, G.; and Viano, L. 2025. Inverse Q-learning done right: Offline imitation learning in Q^π-realizable MDPs. *arXiv preprint arXiv:2505.19946*.

Muslimani, C.; Johnstonbaugh, K.; Chandramouli, S.; Booth, S.; Knox, W. B.; and Taylor, M. E. 2025. Towards improving reward design in RL: A reward alignment metric for RL practitioners. *arXiv preprint arXiv:2503.05996*.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, 278–287.

Ng, A. Y.; and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, 663–670.

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 (NeuIPS)*, 35: 27730–27744.

Pomerleau, D. A. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3: 88–97.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 1889–1897.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; et al. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sikchi, H.; Zheng, Q.; Zhang, A.; and Niekum, S. 2024. Dual RL: Unification and new methods for reinforcement and imitation learning. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

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

Skalse, J. M. V.; Farnik, L.; Motwani, S. R.; Jenner, E.; Gleave, A.; and Abate, A. 2024. STARC: A general framework for quantifying differences between reward functions. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *Proceedings of the 25th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5026–5033.

Wang, Q.; Xiong, J.; Han, L.; Liu, H.; Zhang, T.; et al. 2018. Exponentially weighted imitation learning for batched historical data. *Advances in Neural Information Processing Systems (NeurIPS)*, 6288–6297.

Wulfe, B.; Ellis, L. M.; Mercat, J.; McAllister, R. T.; and Gaidon, A. 2022. Dynamics-aware comparison of learned reward functions. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*.

Zhang, Z.; Xu, T.; Du, X.; Cao, X.; Sun, Y.; and Yu, Y. 2025. Improving reward model gneralization from adversarial process enhanced preferences. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 76414–76435.

Zhu, B.; Jordan, M.; and Jiao, J. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 43037–43067.