Towards Improving Reward Design in RL: A Reward Alignment Metric for RL Practitioners

Calarina Muslimani[1, †], **Kerrick Johnstonbaugh**[2], **Suyog Chandramouli**[3], **Serena Booth**[4], **W. Bradley Knox**[5], **Matthew E. Taylor**[1,6]
[†]musliman@ualberta.ca

[1]**University of Alberta**
[2]**RLCore**
[3]**Princeton University**
[4]**Brown University**
[5]**University of Texas at Austin**
[6]**Alberta Machine Intelligence Institute (Amii)**

Abstract

Reinforcement learning agents are fundamentally limited by the quality of the reward functions they learn from, yet reward design is often overlooked under the assumption that a well-defined reward is readily available. However, in practice, designing rewards is difficult, and even when specified, evaluating their correctness is equally problematic: *how do we know if a reward function is correctly specified?* In our work, we address these challenges by focusing on *reward alignment* — assessing whether a reward function accurately encodes the preferences of a human stakeholder. As a concrete measure of reward alignment, we introduce the Trajectory Alignment Coefficient to quantify the similarity between a human stakeholder's ranking of trajectory distributions and those induced by a given reward function. We show that the Trajectory Alignment Coefficient exhibits desirable properties, such as not requiring access to a ground truth reward, invariance to potential-based reward shaping, and applicability to online RL. Additionally, in an 11–person user study of RL practitioners, we found that access to the Trajectory Alignment Coefficient during reward selection led to statistically significant improvements. Compared to relying only on reward functions, our metric reduced cognitive workload by 1.5x, was preferred by 82% of users and increased the success rate of selecting reward functions that produced performant policies by 41%.

1 Introduction

In reinforcement learning (RL), the *reward hypothesis* states that "all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)" (Sutton & Barto, 2018). More generally, this means that RL agents can solve a task provided the reward function properly defines the task's objective. However, the reward hypothesis does not address the practical challenges of designing reward functions. In practice, reward design is often a difficult and error-prone process carried out by human engineers (Skalse et al., 2022; Booth et al., 2023; Knox & MacGlashan, 2024).

These challenges can become more pronounced in real-world RL applications, where reward design is typically a *collaborative process* between RL practitioner(s) and domain expert(s). While the domain expert has specialized knowledge of the task, they typically lack RL expertise, making it difficult for them to define a reward function explicitly. Instead, the domain expert might express preferences, constraints, or desired outcomes, leaving the RL practitioner responsible for designing

(or selecting) a reward function that satisfies these preferences. This collaboration can increase the complexity of crafting reward functions that correctly specify objectives.

Sparse reward functions are conceptually simple to understand and implement but are less commonly used in practice since current RL algorithms struggle to learn from infrequent signals (Pignatelli et al., 2024). To overcome this, dense reward functions are employed, which provide more frequent feedback to help mitigate the credit assignment problem. However, reward misspecification remains a challenge (Amodei et al., 2016; Skalse et al., 2022). For example, a recent survey found that reward shaping, a method intended to facilitate learning, is commonly used in RL applications for autonomous driving (Knox et al., 2023); without careful design, reward shaping can introduce unintended biases. This can result in RL agents exploiting shortcuts in the reward function or failing to achieve the "true" task objective (Pan et al., 2022). Such issues can pose serious safety risks in real-world applications like autonomous driving and industrial process control.

Reward evaluation is also challenging. This is the process of assessing whether a reward function accurately captures the intended task. A common approach is the "rollout" method, where a policy is trained to optimize the reward function, and then its rollouts are examined to assess the learned behavior (Booth et al., 2023; Gleave et al., 2021). However, this approach has several limitations: (1) it is computationally expensive, (2) can result in reward overfitting—where reward functions become unintentionally over-engineered for a specific algorithm or environment configuration—and (3) assumes that policies are evaluated outside of training, making it less applicable to the online RL setting. Alternatively, prior works (Gleave et al., 2021; Wulfe et al., 2022) have proposed distance metrics for reward evaluation but these require a ground-truth reward for baseline comparison, limiting their integration into the reward design pipeline (unless shaping a reward based on an existing function). Moreover, other metrics (Knox & MacGlashan, 2024; Brown et al., 2021) focus solely on alignment verification and do not measure partial alignment.

In this work, we focus on *reward alignment* as a means of reward evaluation, which we define as the extent to which a reward function preserves human preferences. To operationalize this concept, we introduce the *Trajectory Alignment Coefficient* (σ_{TAC}). This metric evaluates the similarity between a human stakeholder's preferences over trajectory distributions (of which trajectories are a special case) and those induced by a given reward, discount factor pair. It overcomes key limitations of previous work by eliminating the need for a ground-truth reward, instead relying on human preferences. Unlike alignment verification, the Trajectory Alignment Coefficient measures the *degree* of reward alignment, allowing it to distinguish between reward functions that yield the same optimal policy but rank intermediate trajectory distributions differently—making it suitable for online RL.

Additionally, we prove the necessary and sufficient conditions for the Trajectory Alignment Coefficient to be invariant to common transformations, in particular potential-based shaping and positive linear rescaling. This invariance is important because sensitivity to these transformations can cause functionally equivalent rewards to receive different scores, leading to unreliable assessments. Beyond reward alignment, the Trajectory Alignment Coefficient can serve as a distance metric for comparing reward functions (and their associated discounting). While our primary focus is on reward design with human preferences as the reference, this perspective highlights its potential as a tool for comparing reward functions more broadly.

Lastly, we assess whether the Trajectory Alignment Coefficient can aid RL practitioners in the reward design process (see Figure 1). Specifically, we investigate its benefit in reward selection—i.e., choosing performant reward functions that capture a domain expert's preferences. To evaluate this, we conducted an 11–person user study in the Hungry-Thirsty domain (Singh et al., 2009), a test-bed where RL practitioners have struggled to design well-specified rewards (Booth et al., 2023). Our statistically significant findings show that access to the Trajectory Alignment Coefficient during reward selection (1) reduced perceived cognitive workload by $1.5x$, (2) was preferred by 82% of users over the Reward Only condition, and (3) increased the success rate of selecting reward functions that produced performant policies by 41%. Ultimately, our work takes a step toward improving reward

design and selection in RL by introducing a metric that measures the alignment between a proposed reward function and a set of human preferences over sampled trajectory distributions.

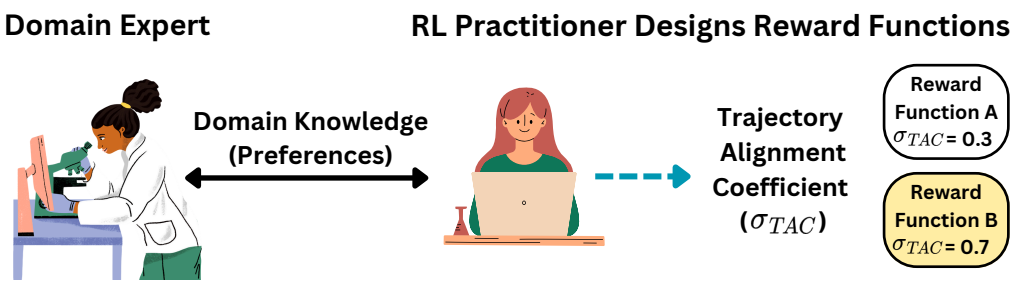


Figure 1: This illustrates the interaction between a domain expert and an RL practitioner in real-world applications (black arrow) and how our metric, σ_{TAC}, integrates into this process (blue arrow).

2 Related Work

Early alignment research focused on directly training agents to align with human preferences (Hadfield-Menell et al., 2016). However, these approaches did not include an assessment of the agent's alignment. Recent efforts have shifted to evaluating the quality of engineered and learned reward functions. For example, Booth et al. (2023); Knox et al. (2023) have conducted empirical investigations to identify shortcomings in current reward design practices and evaluation schemes. Furthermore, metrics have been proposed to compare reward functions without requiring policy evaluations (see Table 1). While some methods are invariant to potential-based shaping, they can rely on access to a ground-truth reward function (Wulfe et al., 2022; Gleave et al., 2021), which is often impractical. Likewise, Brown et al. (2021) proposed verification methods to assess the alignment of an RL agent's behavior but define alignment in terms of optimal policies. We argue, however, that defining alignment in this manner can be limiting, particularly in online RL where one cares about the agent's lifetime performance. Our work builds on Knox & MacGlashan (2024), which described methods to identify misalignment and outlined its common causes. However, this prior work focused on detecting whether misalignment exists, offering only a binary assessment. We extend this framework by introducing a real-valued metric that quantifies the degree of alignment, enabling a more nuanced evaluation of reward function quality. Lastly, while LLM alignment is also a prominent topic (Shen et al., 2023), it is beyond the scope of this work due to its broad focus, which can include mitigating adversarial attacks and detecting bias.

Metric	Invariant	No GT r Required	Not Binary	No Human Preferences	Suitable for Online RL
Gleave et al. (2021)	✓	×	✓	(✓)	✓
Wulfe et al. (2022)	✓	×	✓	(✓)	✓
Brown et al. (2021)	–	×	×	(✓)	×
Knox & MacGlashan (2024)	✓	✓	×	×	×
Trajectory Alignment Coefficient	✓	✓	✓	×	✓

Table 1: Comparison of reward evaluation measures. ✓ indicates the metric satisfies the property, × indicates it does not, (✓) indicates partial satisfaction, and – indicates the property was not evaluated.

3 Background

This section first provides background on RL, then discusses how a reward function and discount factor pair, (r, γ), induce preference orderings over trajectories (and trajectory distributions), a concept rooted in prior work on policy preferences (Bowling et al., 2023).

Definition 1. *A Markov decision process (MDP) is defined by the tuple $(\mathcal{S}, \mathcal{A}, r, p, \mu, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function and $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition function. The initial state distribution is given by μ, and $\gamma \in [0, 1)$ is the discount factor that controls the weighting of future rewards.*

In RL, at every time-step t, the agent takes an action a_t in state s_t, transitions to state s_{t+1}, and then receives reward r_{t+1}. A trajectory τ is a sequence of (s_t, a_t, s_{t+1}) tuples that either reaches a terminal state after a finite number of steps or continues indefinitely. The return of a trajectory is defined as the sum of discounted future rewards, $G_r(\tau) = \sum_{t=0}^{T} \gamma^t r_{t+1}$, where $T = |\tau| - 1$ for episodic tasks or $T \to \infty$ for continuing tasks. The agent attempts to learn a policy, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ to maximize the expected return.

Reward Functions Induce Preference Orderings Consider the deterministic case where we define preferences over a set of trajectories that share the same start state. In this case, given (r, γ) a preferred trajectory is one that yields a greater return:

$$\tau_A \underset{(r,\gamma)}{\succsim} \tau_B \iff G_r(\tau_\text{A}) \geq G_r(\tau_\text{B}) \tag{3.1}$$

where $\tau_A \underset{(r,\gamma)}{\succ} \tau_B$ indicates that trajectory τ_A is preferred over τ_B with respect to (r, γ).

We now shift to the stochastic setting, which arises when the environment or the agent's behavior is stochastic. In this case, we consider probability distributions over trajectories. Note, we specifically focus on trajectory distributions rather than policies, as some distributions (e.g., those that are generated from non-Markovian policies) cannot correspond to any Markov policy.

Definition 2. *Let $H(\mu)$ be the set of all probability distributions over trajectories that share the same initial state distribution μ. That is,*

$$H(\mu) = \{\eta(\tau) \mid \eta(\tau) = \mu(s_0)P(a_0, s_1, a_1, \ldots \mid s_0)\},$$

where $P(a_0, s_1, a_1, \ldots \mid s_0)$ is an arbitrary conditional distribution over trajectories given the initial state s_0. We refer to $\eta(\tau) \in H(\mu)$ as a trajectory distribution and omit the explicit dependence on τ for brevity.

Definition 3. *Given (r, γ), we define a preference ordering over trajectory distributions as follows:*

$$\eta_A \underset{(r,\gamma)}{\succsim} \eta_B \iff \mathbb{E}_{\tau_A \sim \eta_A}[G_r(\tau_A)] \geq \mathbb{E}_{\tau_B \sim \eta_B}[G_r(\tau_B)] \tag{3.2}$$

Equations (3.1) and (3.2) imply that the (r, γ) pair naturally induce a preference ordering over trajectories (or trajectory distributions) via the expected return. To illustrate these concepts, consider the simple autonomous driving task (Knox & MacGlashan, 2024) in Figure 2. Suppose there exists only three trajectories $\{\tau_\text{success}, \tau_\text{idle}, \tau_\text{crash}\}$, and a trajectory distribution $\eta_\text{success-crash}$. τ_success consists of safe driving. τ_crash consists of a car crashing and τ_idle consists of a car remaining parked. $\eta_\text{success-crash}$ is a trajectory distribution that places 90% of its probability mass on τ_success and 10% on τ_crash. Next, consider the pair (r, γ) with return values: $G_r(\tau_\text{success}) = 10$, $G(\tau_\text{idle}) = 0$, $G_r(\tau_\text{crash}) = -50$. By the probabilities of $\eta_\text{success-crash}$, $\mathbb{E}_{\tau \sim \eta_\text{success-crash}}[G_r(\tau)] = 4$. Based on equations (3.1) and (3.2), the resulting preference ordering is $\tau_\text{success} \succ \eta_\text{success-crash} \succ \tau_\text{idle} \succ \tau_\text{crash}$.

4 An Alignment Metric for Reward Function Evaluation

This section introduces the Trajectory Alignment Coefficient as a reward alignment metric and establishes its key theoretical properties.

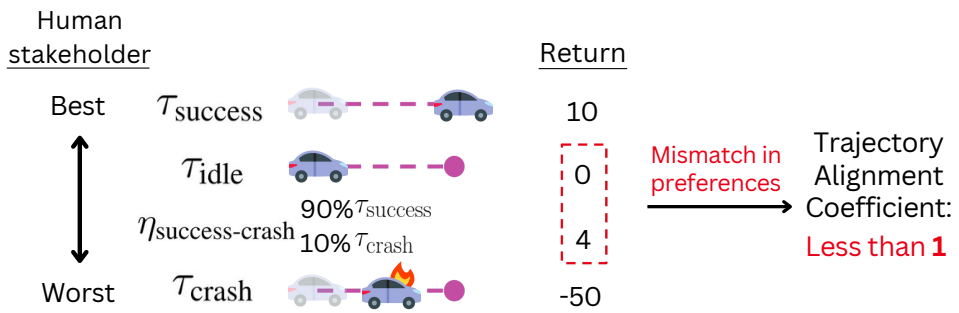


Figure 2: This provides an example of the Trajectory Alignment Coefficient in the case of a simple autonomous driving scenario.

4.1 Trajectory Alignment Coefficient

To establish a reward alignment metric, we need to quantify how well a reward function reflects the preferences of a human stakeholder. To achieve this, we propose the *Trajectory Alignment Coefficient*, a measure based on Kendall's Tau–b correlation. Kendall's Tau–b is a non-parametric measure that quantifies the level of agreement between two sets of ranked data, adjusting for ties (Kendall, 1945). It outputs a scalar value $\in [-1, 1]$, indicating levels of agreement: 1 for perfect agreement (e.g., identical preference orderings) and -1 for complete disagreement (e.g., reverse preference orderings). The Trajectory Alignment Coefficient measures the similarity among preference orderings over trajectory distributions in $H(\mu)$. However, $H(\mu)$ can theoretically contain an intractably large number of trajectory distributions. To apply the Trajectory Alignment Coefficient as a practical reward alignment measure, we must consider finite subsets of trajectory distributions from $H(\mu)$.

To compute σ_{TAC}, we first construct two preference data sets, one from a human (D_h), assumed to be transitive, and one induced by a given (r, γ) pair. Specifically, we define:

$$\upsilon_h(D_h) = \Big\{ \{\eta_i, \eta_j\} \mid (\eta_i \underset{(h)}{\diamond} \eta_j) \in D_h \Big\}$$

where $\diamond \in \{\succ, \prec, \sim\}$ denotes a preference relation and the subscript indicates whether the preference originates from the human or (r, γ). υ extracts unordered pairs of trajectory distributions that were ranked by the human. Then given these pairs, we construct the corresponding preference dataset under (r, γ) via Definition (3), which ranks trajectory distributions with respect to (r, γ):

$$D_{r,\gamma}(\upsilon_h, r, \gamma) = \Big\{ (\eta_i \underset{(r,\gamma)}{\diamond} \eta_j) \mid \{\eta_i, \eta_j\} \in \upsilon_h(D_h) \Big\}.$$

Once we have both D_h and $D_{r,\gamma}$, σ_{TAC} measures their agreement using Kendall's Tau-b:

$$\sigma_{TAC}(D_h, D_{r,\gamma}) \doteq \frac{P - Q}{\sqrt{(P + Q + X_0)(P + Q + Y_0)}} \tag{4.1}$$

where

$$\begin{aligned} P &: \text{Number of concordant pairs between } D_{r,\gamma} \text{ and } D_h, \\ Q &: \text{Number of discordant pairs between } D_{r,\gamma} \text{ and } D_h, \\ X_0 &: \text{Number of pairs tied only in } D_{r,\gamma}, \\ Y_0 &: \text{Number of pairs tied only in } D_h. \end{aligned}$$

This formulation ensures that σ_{TAC} quantifies the alignment between human and reward-induced preferences over the same set of trajectory distribution pairs. To illustrate, consider the trajectory

distributions, η_1, η_2, η_3, where $D_h, D_{r,\gamma}$ are as follows,

$$D_h = \Big\{(\eta_2 \underset{(h)}{\succ} \eta_3), (\eta_1 \underset{(h)}{\succ} \eta_3)\Big\} \implies \upsilon_h(D_h) = \Big\{\{\eta_2, \eta_3\}, \{\eta_1, \eta_3\}\Big\}$$

$$D_{r,\gamma}(\upsilon_h, r, \gamma) = \Big\{(\eta_2 \underset{(r,\gamma)}{\prec} \eta_3), (\eta_1 \underset{(r,\gamma)}{\succ} \eta_3)\Big\}.$$

Note that given a subset of trajectory distributions, the Trajectory Alignment Coefficient can be applied in cases with either a full or partial ranking. A full ranking establishes a complete order over the elements in a subset, where all necessary pairwise comparisons are available. In contrast, a partial ranking occurs when some pairwise comparisons are missing (e.g., $D_{r,\gamma}, D_h$, as the comparison between η_1 and η_2 is missing). This flexibility allows the Trajectory Alignment Coefficient to be used in settings where ranking information is limited or incomplete.

Moreover, the definition for σ_{TAC} in Equation (4.1) can also be used to evaluate the differences between any two reward, discount factor pairs: $(r, \gamma), (r', \gamma)$. For example, given data sets $D_{r,\gamma}$ and $D_{r',\gamma}$ representing preferences over trajectory distributions, we can use Equation (4.1) to determine how similar the preferences induced by (r, γ) are to those induced by (r', γ).

4.2 Invariance to Common Reward Transformations

Reward shaping is commonly used to accelerate RL training by modifying the reward function to improve learning efficiency (Ng et al., 1999). Common reward transformations include potential-based reward shaping and positive linear rescaling. A well-designed reward evaluation metric should be invariant to these transformations; otherwise, it may assign different scores to functionally equivalent rewards, leading to inconsistent assessments. We show that the Trajectory Alignment Coefficient maintains this invariance, ensuring stable evaluations of reward alignment.

Definition 4. *A potential-based reward function is defined as* $r'(s, a, s') \doteq r(s, a, s') + \gamma\Phi(s') - \Phi(s)$, *given a potential function* $\Phi : \mathcal{S} \to \mathbb{R}$, *and* γ *as the MDP discount factor.*

To determine whether the Trajectory Alignment Coefficient is invariant to potential-based reward shaping, we first examine the conditions for which preference orderings remain unchanged. While potential-based reward shaping is known to preserve the optimal policy (Ng et al., 1999), we further prove in Theorem 4.4 that, in the infinite-horizon setting, it preserves preference orderings over all trajectory distributions $\eta \in H(\mu)$ *if and only if they share the same start-state distribution,* μ. This establishes a fundamental condition for ensuring that any reward alignment metric based on preference orderings remains unaffected by potential-based reward shaping.

Lemma 4.1. *Given the infinite-horizon setting, if the expected returns under reward function* r' *are a positive linear transformation of the expected returns under reward function* r, *with respect to all trajectory distributions, then the preference ordering over any two trajectory distributions* η_i *and* η_j *remains unchanged. Formally:*

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \alpha\mathbb{E}_{\tau\sim\eta}[G_r(\tau)] + \beta \implies \Big(\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \iff \eta_i \underset{(r',\gamma)}{\succsim} \eta_j\Big) \quad \forall \eta_i, \eta_j,$$

where $\alpha > 0$ *and* β *are constants and the expectations* $\mathbb{E}_{\tau\sim\eta}[G_r(\tau)]$ *and* $\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)]$ *are taken over the same trajectory distributions.*

Lemma 4.2. *[Sufficiency] In the infinite horizon setting, if two trajectory distributions* $\eta_i, \eta_j \in H(\mu)$, *then potential-based reward shaping preserves their preference ordering with respect to the reward function,* r, *and the potential-based function,* r'*:*

$$\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \iff \eta_i \underset{(r',\gamma)}{\succsim} \eta_j.$$

Proof. Let $\eta_i, \eta_j \in H(\mu)$ be arbitrary trajectory distributions, and without loss of generality assume that $\eta_i \underset{(r,\gamma)}{\succsim} \eta_j$. From Definition (3), this implies that $\mathbb{E}_{\tau\sim\eta_i}[G_r(\tau)] \geq \mathbb{E}_{\tau\sim\eta_j}[G_r(\tau)]$. We now

analyze how the expected return changes under the potential-based reward function r'. The expected return under the reward function r and the shaped reward function r' is:

$$\mathbb{E}_{\tau\sim\eta}[G_r(\tau)] = \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t, s_{t+1})\right], \tag{4.2}$$

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t r'(s_t, a_t, s_{t+1})\right] \tag{4.3}$$

Substitute the definition of the potential-based reward function, Definition (4), into Equation (4.3):

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t\Big(r(s_t, a_t, s_{t+1}) + \gamma\Phi(s_{t+1}) - \Phi(s_t)\Big)\right]$$

Distribute γ^t and rearrange terms:

$$\begin{aligned}
\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] &= \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t, s_{t+1}) + \gamma^t\gamma\Phi(s_{t+1}) - \gamma^t\Phi(s_t)\right] \\
&= \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t, s_{t+1})\right] + \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^{t+1}\Phi(s_{t+1}) - \gamma^t\Phi(s_t)\right] \\
&= \mathbb{E}_{\tau\sim\eta}[G_r(\tau)] + \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=1}^{\infty}\gamma^t\Phi(s_t) - \sum_{t=0}^{\infty}\gamma^t\Phi(s_t)\right]
\end{aligned}$$

Split $\sum_{t=0}^{\infty}\gamma^t\Phi(s_t)$ into two parts, one from $t = 1$ to ∞ and the other isolating $t = 0$:

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \mathbb{E}_{\tau\sim\eta}[G_r(\tau)] + \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=1}^{\infty}\gamma^t\Phi(s_t) - \sum_{t=1}^{\infty}\gamma^t\Phi(s_t) - \gamma^0\Phi(s_0)\right]$$

Now, combine like-terms and $\sum_{t=1}^{\infty}\gamma^t\Phi(s_t)$ gets canceled out:

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \mathbb{E}_{\tau\sim\eta}[G_r(\tau)] - \mathbb{E}_{\tau\sim\eta}\left[\Phi(s_0)\right]$$

As $\Phi(s_0)$ depends only on the start-state distribution μ, and μ is the same for all $\eta \in H(\mu)$, we conclude that the expected returns under r and r' differ by a constant, $\mathbb{E}_{s_0\sim\mu}\left[\Phi(s_0)\right]$:

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \mathbb{E}_{\tau\sim\eta}[G_r(\tau)] - \mathbb{E}_{s_0\sim\mu}\left[\Phi(s_0)\right] \tag{4.4}$$

As $\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)]$ is a positive linear transformation of $\mathbb{E}_{\tau\sim\eta}[G_r(\tau)]$, we apply Lemma 4.1 and conclude that the preference remains unchanged under reward shaping:

$$\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \iff \eta_i \underset{(r',\gamma)}{\succsim} \eta_j \;\forall \eta_i, \eta_j \in H(\mu)$$

□

Lemma 4.3. *[Necessity] In the infinite horizon setting, if two trajectory distributions $\eta_i \in H(\mu_i)$ and $\eta_j \in H(\mu_j)$ have different start-state distributions ($\mu_i \neq \mu_j$), then there exists a potential function Φ such that:*

$$\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \text{ and } \eta_i \underset{(r',\gamma)}{\prec} \eta_j.$$

We leave the proof for the necessity condition (Lemma (4.3) to Supplementary Material C.

Theorem 4.4. *In the infinite-horizon setting, let (r, γ) and two trajectory distributions $\eta_i \in H(\mu_i)$ and $\eta_j \in H(\mu_j)$ be given. Potential-based reward shaping is guaranteed to maintain the preference ordering over all trajectory distributions if and only if $\mu_i = \mu_j$. Formally,*

$$\big(\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \iff \eta_i \underset{(r',\gamma)}{\succsim} \eta_j\big)\ \forall \Phi \in \mathcal{F} \iff \mu_i = \mu_j$$

where $\mathcal{F}$ is the space of all potential-based shaping functions and $\Phi : \mathcal{S} \mapsto \mathbb{R}$ is an arbitrary function in this space. The proof of Theorem (4.4) follows directly from Lemmas (4.2) and (4.3).

Definition 5. *Let $D_{r,\gamma}$, $D_{r',\gamma}$, and D_h be preference data sets over trajectory distributions induced by (r, γ), (r', γ), and a human, respectively, where $r' \doteq f(r)$ for some transformation f. The Trajectory Alignment Coefficient σ_{TAC} is invariant to f if and only if $\sigma_{TAC}(D_h, D_{r,\gamma}) = \sigma_{TAC}(D_h, D_{r',\gamma})$.*

Theorem 4.5. *Consider the infinite-horizon setting. Let r and r' be reward functions where r' is a shaped version of r using potential-based reward shaping. Let D_h be a data set of human preferences over trajectory distributions, and define $D_{r,\gamma} = D_{r,\gamma}(\upsilon_h, r, \gamma)$ and $D_{r',\gamma} = D_{r',\gamma}(\upsilon_h, r', \gamma)$ as the preference data sets induced by (r, γ) and (r', γ), respectively. The Trajectory Alignment Coefficient is invariant to any potential-based reward shaping if and only if within each set of trajectory distributions $\{\eta_i, \eta_j\} \in \upsilon_h(D_h)$, η_i, η_j share the same initial state distribution (i.e., $\eta_i, \eta_j \in H(\mu)$).*

Proof. Let $\{\eta_i, \eta_j\} \in \upsilon_h(D_h)$ be an arbitrary pair of trajectory distributions compared in the human preference data set. By Definition (4.1) of σ_{TAC}, the following biconditional holds:

$$\sigma_{\text{TAC}}(D_h, D_{r,\gamma}) = \sigma_{\text{TAC}}(D_h, D_{r',\gamma}) \iff \forall \{\eta_i, \eta_j\} \in \upsilon_h(D_h) : \big(\eta_i \underset{(r,\gamma)}{\diamond} \eta_j \iff \eta_i \underset{(r',\gamma)}{\diamond} \eta_j\big)$$

where $\diamond \in \{\succ, \prec, \sim\}$ denotes the preference relation. By Theorem (4.4), we have:

$$\big(\eta_i \underset{(r,\gamma)}{\diamond} \eta_j \iff \eta_i \underset{(r',\gamma)}{\diamond} \eta_j\big)\ \forall \Phi \in \mathcal{F}, \eta_i \in H(\mu_i), \eta_j \in H(\mu_j) \iff \mu_i = \mu_j$$

Therefore, we conclude that σ_{TAC} is invariant to potential-based reward shaping if and only if all sets of trajectory distributions being compared share the same initial state distribution, μ:

$$\sigma_{\text{TAC}}(D_h, r, \gamma) = \sigma_{\text{TAC}}(D_h, r', \gamma)\ \forall \Phi \in \mathcal{F} \iff \eta_i, \eta_j \in H(\mu),\ \forall \{\eta_i, \eta_j\} \in \upsilon_h(D_h)$$

□

Theorem 4.6. *Given the infinite-horizon setting, the Trajectory Alignment Coefficient is invariant to positive linear transformations.*

To prove Theorem (4.6), we show that a positive linear transformation linearly transforms the expected return. We can then apply Lemma (4.1) and the Trajectory Alignment Coefficient's invariance definition (5) to complete the proof. The full derivation is provided in Supplementary Material C.

4.3 Trajectory Alignment Coefficient in Practice

To use the Trajectory Alignment Coefficient in practice, two parameters must be specified: the number of trajectories (or trajectory distributions) to rank and the trajectory sampling method. To obtain a meaningful evaluation, it is important to include a diverse set of trajectories. If trajectories are limited to a specific region of the state and action space, the evaluation may fail to reveal reward misalignment in other regions. To mitigate this, one heuristic we propose is to sample trajectories that exhibit qualitatively different behaviors. In our experiments, we generate these trajectories via RL agents partially trained with different reward functions. However, other sources, such as human demonstrations or trajectories generated through behavioral cloning, could also be used. Beyond diversity, the number of ranked trajectories plays a critical role. While more trajectories provide a clearer picture of reward alignment, ranking too many is impractical for humans. We found that in

our tested domain (Hungry-Thirsty, described in Section 5), the Trajectory Alignment Coefficients from a subset of 12 trajectories were highly correlated with those from a set of 1200, suggesting that a smaller, well-chosen set can still provide reliable estimates. See Supplementary Material A.2 for further details.

To illustrate the Trajectory Alignment Coefficient in practice, we revisit the toy example in Figure 2, demonstrating how reward-based rankings can diverge from human preferences Specifically, the toy (r, γ) pair produced a preference ordering of $\tau_{\text{success}} \succ \eta_{\text{success-crash}} \succ \tau_{\text{idle}} \succ \tau_{\text{crash}}$. However, a human stakeholder would likely prefer remaining parked over possibly crashing: $\tau_{\text{success}} \succ \tau_{\text{idle}} \succ \eta_{\text{success-crash}} \succ \tau_{crash}$. To compute σ_{TAC} from Equation (4.1), we count the number of concordant and discordant pairs. In our four-element example, six pairwise comparisons are possible, with all but one being concordant. The only discordant pair is $(\eta_{\text{success-crash}}, \tau_{\text{idle}})$, where the preference is reversed. Since there are no ties, X_0 and Y_0 are zero. Substituting these values into the equation yields $\sigma_{TAC} \approx 0.67$, indicating misalignment between (r, γ) and the human stakeholder's preferences.

5 Experimental Design

This study examines the reward design setting where RL practitioners have to choose between reward functions in order to satisfy the preferences of another stakeholder (e.g., a domain expert). In particular, *our goal is to investigate whether the Trajectory Alignment Coefficient can assist RL practitioners in reward selection*. We assess this by comparing RL practitioners with and without access to our metric, focusing on two key dimensions:

1. **Perceived Benefit** — Does it reduce perceived cognitive workload, increase ease of use, and improve understanding of reward functions?
2. **Practical Impact** — Does access to the metric help RL practitioners choose reward functions that improve performance of learned policies while also reducing the time spent on reward selection?

We conducted an ethics-approved human subject study with 11 self-identified RL practitioners. This included individuals who had completed graduate courses in RL (81%), conducted RL research (100%), or applied RL in their professional work (27%). Note these categories were not mutually exclusive. In this within-subjects study, participants selected reward functions under three experimental conditions with different types of assistance. The study was primarily in-person and each session lasted approximately 90 minutes.

Testbed: Hungry-Thirsty Our study is conducted in a modified Hungry-Thirsty domain (Singh et al., 2009), an environment where others have shown (Booth et al., 2023) that RL practitioners struggle with reward design. It is a 4×4 grid-world where food and water are randomly placed at the grid corners (see Figure 12 in Supplementary Material D). The agent can move in one of the four cardinal directions or execute eat or drink actions. The agent's goal is to maximize time spent without hunger. Hunger occurs if the agent has not eaten in the previous timestep, but eating is only possible at a food source when the agent is not thirsty. If the agent is thirsty, the eat action fails. The agent becomes thirsty with 0.10 probability per step. This is an infinite-horizon MDP, although each episode is truncated after 200 timesteps. The state space consists of the agent's position and two Boolean variables for hunger and thirst. The reward function is a linear combination of these variables. The evaluation metric is the number of timesteps the agent is not hungry.

Study Protocol The study consisted of two primary components, Preference Review and Reward Selection. In the *Preference Review* component, participants first read a description of the Hungry-Thirsty domain and then completed a short quiz and an interactive game-play session to confirm their understanding of its rules. They were informed that they would be collaborating with a domain expert to select a reward function for training an RL agent, with the expert providing a ranking of 12 trajectories (generated using the task evaluation metric as a proxy for expert preferences). Participants then reviewed this ranking alongside corresponding video clips. To obtain these trajectories, we use the mixture sampling method described in Section 4.3.

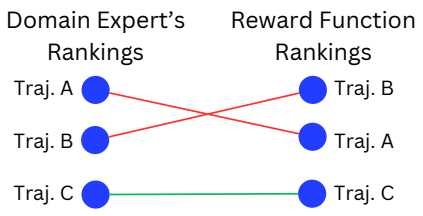


Figure 3: Visualization comparing rankings of 3 trajectories, used in the Reward + Alignment + Visualization condition.

In the *Reward Selection phase*, participants completed four rounds of reward selection, with the goal of choosing the reward function that best reflects the domain expert's preferences. To select reward functions for this component, we considered those from the open-sourced human reward data set in Booth et al. (2023) and their affine transformations. See Table 4 in Supplementary Material B for the complete set of reward function comparisons. When selecting reward functions for comparisons, we focused on reward functions that: (1) differed in magnitude, scale, or range, and (2) myopically ranked states based on immediate reward. Across all conditions, participants had access to two reward functions at a time and could revisit the trajectory rankings along with their respective video clips. Moreover, reward function pairs were randomized and were not repeated across the study. See Figures 6–9 in Supplementary Material B for the user interface. The study contained three conditions, whose order was randomized for each participant:

- **Reward Only (Control):** The reward functions were shown but no further information was given.
- **Reward + Alignment:** Participants also received the Trajectory Alignment Coefficient, computed from the domain expert's preferences and those induced by each reward function.
- **Reward + Alignment + Visual:** Participants were also provided the Trajectory Alignment Coefficient and a parallel coordinate plot illustrating differences in the domain expert's and reward functions' rankings over trajectories (see example in Figure 3).

Evaluation To assess differences in *perceived benefit* across conditions, participants completed a modified NASA Task Load Index (NASA-TLX) (Hart & Staveland, 1988), which measured cognitive work load (scale from 1–7). The survey included all six NASA-TLX questions, along with three additional questions on confidence in reward selection, helpfulness of feedback (e.g., alignment, visual, and/or reward function), and ease of integrating feedback into decisions (see Figure 10 in Supplementary Material B). Participants completed this survey after each condition. We then compute the overall workload based on the survey responses. After completing all reward selection conditions, participants voted on which condition best improved their understanding of the reward function, provided the most useful feedback, was least mentally demanding, and made reward selection the easiest (see Figure 11 in Supplementary Material B). We also included open-ended questions for participants to describe their experiences during the reward selection process.

To assess the *practical impact* of the conditions, we first examined how often users selected reward functions that improved policy performance compared to the unselected alternative rewards. For shorthand, we refer to these reward functions as performant or policy-improving. Specifically, we calculated the proportion of times users chose the reward function that resulted in a higher final return and a greater area under the learning curve (AUC), with respect to the evaluation metric. To evaluate the policy performance of the reward functions, we trained Q–Learning, SARSA, and Expected SARSA agents on each reward function, performing a grid search over learning rate $\in \{10^{-k}, 5 \times 10^{-k} \mid k \in \{2, 3, 4\}\}$, and epsilon $\in \{0.01, 0.05, 0.15\}$. Each agent was trained across 10 environment seeds. We then averaged final returns and AUC, separately, across all trained agents to assess performance. By using multiple RL algorithms and varying hyperparameters, we aimed to reduce the likelihood that a reward function's performance was due to random chance or a particularly favorable choice of hyperparameters. Second, we measured the time taken to complete the reward selection process.

For all analyses, we used paired t–tests for continuous data when normality assumptions held and Wilcoxon Signed-Rank tests otherwise. For categorical voting data, Fisher's Exact Test was applied.

The corresponding p–values and test statistics are reported: t (paired t–test) and W (Wilcoxon Signed-Rank test). We performed the Bonferroni correction ($\alpha = 0.05$) to control for Type I errors.

6 Results

This section presents the results from the user study, structured around our two research components, perceived benefit and practical impact, as well as a qualitative analysis of participants' experiences.

Perceived Benefit Figure 4(a) presents the results of the NASA-TLX survey: the average ratings ($\pm$ standard error) for five selected questions and the overall workload score. We chose a subset of questions for this plot to avoid redundancy. We found that the overall workload score was significantly lower for the Reward + Alignment ($\mu = 1.97$, $p = 0.003$, $t = -3.46$) and Reward + Alignment + Visual ($\mu = 1.83$, $p = 0.02$, $W = 10.0$) conditions compared to the Reward Only condition ($\mu = 3.19$), representing a $1.5x$ reduction in workload. Similarly, 4(b) shows the total number of participant votes for each condition, reflecting preferences across different criteria. Notably, 100% of participants reported that either alignment conditions led to easier decision-making. Additionally, 91% indicated that either alignment conditions improved their understanding of the reward functions and reduced mental demand, while 82% found the provided information most useful—all of which are significantly greater than the number of votes for the Reward Only condition ($p \leq 0.009$).

Practical Impact In Figure 5(b), we found that participants achieved significantly greater success in selecting the policy-improving reward functions in both the Reward + Alignment ($\mu = 0.93$, $p = 0.01$, $W = 41.0$) and Reward + Alignment + Visual ($\mu = 0.96$, $p = 0.008$, $W = 28.0$) conditions compared to the Reward Only condition ($\mu = 0.66$). Specifically, we found that in the Reward Only condition, 55% of participants selected policy-improving reward functions no better than random (or worse). However, the time-to-completion data, shown in Figure 5(a), provide a more complete picture. While participants on average took longer in the Reward Only condition ($\mu = 660.89$) compared to both the Reward + Alignment ($\mu = 334.09$, $p = 0.04$, $W = 13.0$) and Reward + Alignment + Visual conditions ($\mu = 393.44$, $p = 0.07$, $W = 16.0$), these differences were not statistical significant. It is important to acknowledge that deriving the preference ordering used in the alignment conditions itself requires time, which is not accounted for in this comparison.

Beyond aggregate trends, individual differences in time use revealed interesting patterns. Notably, six participants (P6–P11) spent more time in the Reward Only condition but still performed worse in reward selection, highlighting that more time without alignment support did not lead to better outcomes. Moreover, three participants (P1–P3) achieved perfect success rates in the Reward Only condition while spending less time than in the Reward + Alignment + Visual condition. This suggests that they may have required less assistance during reward selection, and the additional visual feedback in the Reward + Alignment + Visual condition likely introduced more information for them to process, increasing deliberation time. These results suggest that while alignment-based feedback improved reward selection success for most participants, some succeeded without it.

Qualitative Analysis In the open-ended questions, we asked participants to explain which condition they liked and disliked. Most participants favored the Reward + Alignment + Visual condition (73%). A common theme among these participants was the emphasis on how the combination of the visualization and alignment score provided both intuitive insights into the reward function's behavior and a scalar metric that simplified decision-making. For example, **P5** stated "It also let me see exactly which trajectories were aligned vs not, giving me better insights into what behavior the reward function was favoring." Furthermore, the Reward Only condition was least favored by 64% of participants. Two main complaints emerged: (1) difficulty interpreting the reward functions, requiring intuition or domain expertise, and (2) the process being slow and tedious, making reward comparisons time-consuming. Specifically, **P2** wrote "it is very hard to just look at the reward function and guess what will happen," while **P8** mentioned "Even though all the information needed to make a decision could be deducted from the trajectories, going through all of them on a piece of paper could

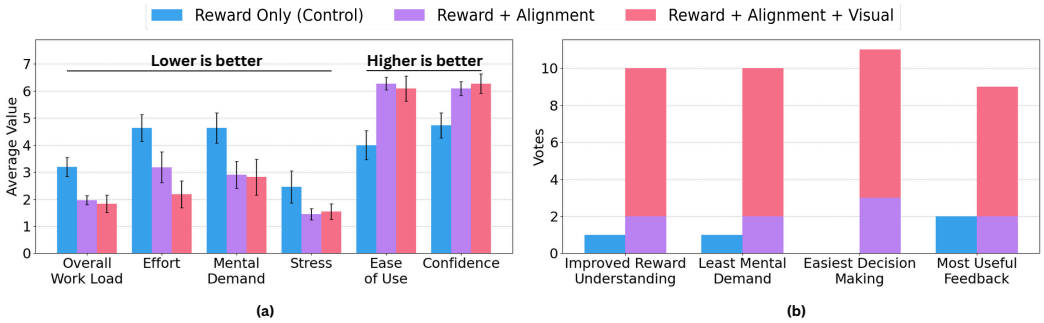


Figure 4: Results from participants' experience during reward selection are shown from (a) participants given the NASA-TLX survey and (b) a survey assessing different aspects of favorability.

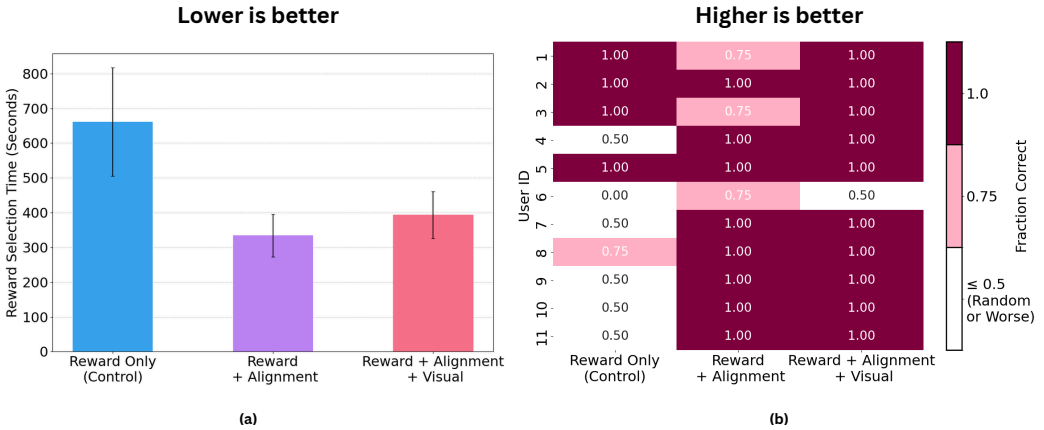


Figure 5: Results showing (a) the mean completion time ($\pm$ standard error) for reward selection and (b) the proportion of policy-improving reward functions selected per user and condition.

be very hard and time-consuming." Lastly, two participants least preferred the Alignment + Reward condition. The common theme was that the alignment score alone was insufficient. Unsurprisingly, participants felt that, compared to the Reward + Alignment + Visual condition, it lacked detailed feedback and appeared too aggregated without supporting visuals, making decision-making more difficult. Overall, the open-ended responses further supported the quantitative evidence, indicating that the Trajectory Alignment Coefficient improved the user experience during reward selection.

7 Conclusion

The success of RL agents is inherently dependent on the quality of the MDP's reward function, yet reward design is often treated as a secondary concern. In practice, however, it is a complex and error-prone process. These challenges are further amplified in real-world RL applications, where reward design is typically a collaborative effort between RL practitioners and domain experts. In such settings, the RL practitioner must design a reward function that accurately reflects the domain expert's preferences and constraints. In this work, we address this challenge by introducing the Trajectory Alignment Coefficient, a reward alignment metric that quantifies the similarity between a human stakeholder's preference orderings over trajectory distributions and those induced by a reward function. Through an 11–person user study, we demonstrate its effectiveness in supporting RL practitioners during reward selection. Specifically, participants in Trajectory Alignment-based conditions reported significantly lower cognitive workload and were more likely to select policy-improving reward functions. In future work, we plan to extend our metric to full reward design, where participants construct reward functions from scratch.

Broader Impact Statement

In this work, we introduce a reward evaluation metric that measures the alignment between a human stakeholder's preferences over trajectory distributions and those induced by a reward function. However, if the preferences provided by the human stakeholder do not accurately reflect their true beliefs, the output of the metric may be unreliable and could mislead the reward design process.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete Problems in AI Safety. *CoRR, abs/1606.06565*, 2016.

Serena Booth, W. Bradley Knox, Julie Shah, Scott Niekum, Peter Stone, and Alessandro Allievi. The Perils of Trial-and-Error Reward Design: Misdesign through Overfitting and Invalid Task Specifications. In *AAAI Conference on Artificial Intelligence*, 2023.

Michael Bowling, John D. Martin, David Abel, and Will Dabney. Settling the Reward Hypothesis. In *International Conference on Machine Learning*, 2023.

Daniel S. Brown, Jordan Schneider, and Scott Niekum. Value Alignment Verification. In *International Conference on Machine Learning*, 2021.

Adam Gleave, Michael Dennis, Shane Legg, Stuart Russell, and Jan Leike. Quantifying Differences in Reward Functions. In *International Conference on Learning Representations*, 2021.

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative Inverse Reinforcement Learning. In *Neural Information Processing Systems*, 2016.

Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, 52:139–183, 1988.

Maurice George Kendall. The Treatment of Ties in Ranking Problems. *Biometrika*, 33:239–251, 1945.

W. Bradley Knox and James MacGlashan. How to Specify Reinforcement Learning Objectives. In *Finding the Frame: An RLC Workshop for Examining Conceptual Frameworks at the Reinforcement Learning Conference*, 2024.

W. Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (Mis)design for Autonomous Driving. *Artificial Intelligence*, 316(103829), 2023.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *International Conference on Machine Learning*, 1999.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models. In *International Conference on Learning Representations*, 2022.

Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. A Survey of Temporal Credit Assignment in Deep Reinforcement Learning. *Transactions on Machine Learning Research*, 2024.

Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. Large Language Model Alignment: A Survey. *CoRR, abs/2309.15025*, 2023.

Satinder Singh, Richard L Lewis, and Andrew G Barto. Where Do Rewards Come From? In *Conference of the Cognitive Science Society*, 2009.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and Characterizing Reward Gaming. In *Neural Information Processing Systems*, 2022.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An Introduction*. MIT press, 2018.

Blake Wulfe, Ashwin Balakrishna, Logan Ellis, Jean Mercat, Rowan McAllister, and Adrien Gaidon. Dynamics-Aware Comparison of Learned Reward Functions. In *International Conference on Learning Representations*, 2022.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Trajectory Alignment Coefficient in Practice

A.1 How to sample trajectories?

To sample the trajectories used for the trajectory alignment coefficient, we propose that practitioners select qualitatively different trajectories. We now outline the methodology used to obtain these trajectories.

To ensure qualitative diversity, we sampled trajectories from Q–learning agents that were only partially trained. Note that for this component, we only considered the environment configuration with a fixed start state of $(0, 0)$ and where the food and water locations are $(3, 0)$ and $(0, 0)$, respectively. We did this intentionally, as the trajectory alignment coefficient only remains invariant to potential-based reward shaping if the start state is fixed.

We used the default hyperparameters from Table 3 and the evaluation metric as the reward function for this training. We categorized partial training into three groups: **low-return**, **medium-return**, and **high-return**. After partial training, we performed offline evaluation (i.e., policy rollouts with no exploration). The low-return group contained trajectories with returns in $[1, 30)$, the medium-return group had returns in $[30, 60)$, and the high-return group had returns ≥ 60. We then randomly sampled four trajectories per group, resulting in a total of 12 trajectories used in the user study outlined in Section 5.

The specific returns (per the evaluation metric) of the 12 trajectories are as follows:

$$[1.0, 6.0, 9.0, 29.0, 43.0, 56.0, 56.0, 66.0, 68.0, 74.0, 90.0]$$

Note that the optimal policy achieves an average return of ≈ 96.31. We computed the optimal policy by performing value iteration over 13 seeds.

A.2 How many trajectories to samples?

In Section 4.3, we noted that in Hungry Thirsty, the Trajectory Alignment Coefficients computed from a small subset of 12 trajectories were highly correlated with those computed from a larger set of 1200 trajectories. To better understand this relationship, we now outline the methodology used to obtain this result.

More specifically, we used all 31 reward functions from the open-sourced human reward data set from Booth et al. (2023), along with their variants (e.g., 23 added reward functions, linear transformations), resulting in a total of 54 reward functions. We then sampled trajectory subsets of varying sizes ($N \in \{10, 12, 25, 100, 500\}$) using the sampling strategy described earlier. Using the task evaluation metric as a proxy for the domain expert preferences, we calculated the Trajectory Alignment Coefficients (σ_{TAC}) between the domain expert's preferences and those induced by the reward functions. To assess whether smaller trajectory subsets provide reliable σ_{TAC} estimates, we computed the correlation between the σ_{TAC} scores from each subset to those obtained using a larger trajectory set of 1200. We repeated this process 50 times per trajectory subset size to account for variability and then averaged the resulting correlations. This is depicted in Table 2.

A high correlation between the Trajectory Alignment Coefficients from smaller subsets and the 1200–trajectory set would indicate that even with a limited number of trajectories, we can obtain σ_{TAC} estimates that are consistent with those derived from a significantly larger sample. This finding suggests that in Hungry-Thirsty, a relatively small number of trajectories may be sufficient for accurately assessing σ_{TAC}, reducing the need for extensive trajectory ranking.

We used *Kendall's Tau* to measure correlation because (1) the normality assumption was violated, making Pearson's r unsuitable, and (2) there were ties in the dataset, which makes Kendall's Tau a better choice than *Spearman's Rho*, as it handles tied ranks more effectively.

SUBSET SIZE	AVERAGE CORRELATION	STANDARD DEVIATION
500	0.992	0.005
100	0.979	0.009
25	0.930	0.036
12	0.828	0.105
10	0.795	0.131

Table 2: Average correlation (across 50 samples) between the σ_{TAC} scores computed from each subset size and those obtained using a larger trajectory set of 1200 (denoted in bold). The standard deviation across samples is reported in the third column.

A.3 Q-Learning Hyperparameters

An overview of the hyperparameters used for training the Q–Learning, SARSA, and Expected SARSA algorithms are provided in Table 3. To evaluate the performance of the reward functions used in the reward selection aspect of the user study, we trained 18 Q–Learning, SARSA and Expected SARSA agents by performing a full grid search over two hyperparameters: learning rate and epsilon. We systematically varied both across all combinations while keeping the remaining hyperparameters fixed.

HYPERPARAMETER	VALUE
NUMBER OF TRAINING EPISODES	10000
NUMBER OF SEEDS	10
LEARNING RATE	[0.0001, 0.001, 0.01, 0.0005, 0.005, **0.05**]
EXPLORATION STRATEGY	EPSILON-GREEDY
EPSILON	[0.05, 0.10,**0.15**]
DISCOUNT	0.99

Table 3: Hyperparameters for all RL Algorithms. For hyperparameters with multiple options, the list represents possible values that were searched over. Bolded values indicate default settings.

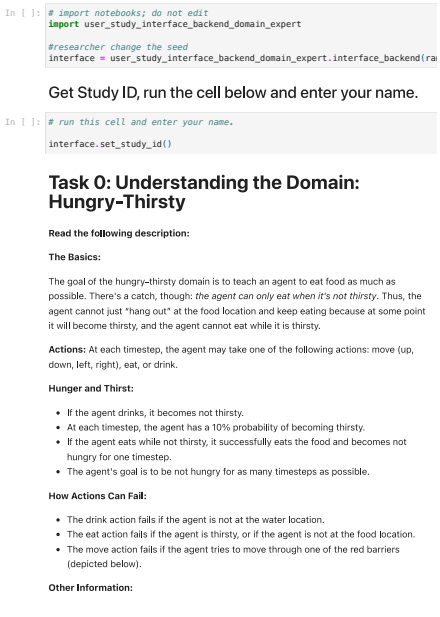
B User Study and Reproducibility

B.1 Code

We open-source our code and the corresponding data from the human subject study at the following link: https://github.com/cmuslima/TAC_RewardAlignment_RLC2025.

B.2 User Study Figures

We first show figures that correspond to the interface used in the human-subject study.



In [ ]:
```
# import notebooks; do not edit
import user_study_interface_backend_domain_expert

#researcher change the seed
interface = user_study_interface_backend_domain_expert.interface_backend(ra
```

Get Study ID, run the cell below and enter your name.

In [ ]:
```
# run this cell and enter your name.

interface.set_study_id()
```

Task 0: Understanding the Domain: Hungry-Thirsty

Read the following description:

The Basics:

The goal of the hungry–thirsty domain is to teach an agent to eat food as much as possible. There's a catch, though: *the agent can only eat when it's not thirsty*. Thus, the agent cannot just "hang out" at the food location and keep eating because at some point it will become thirsty, and the agent cannot eat while it is thirsty.

Actions: At each timestep, the agent may take one of the following actions: move (up, down, left, right), eat, or drink.

Hunger and Thirst:

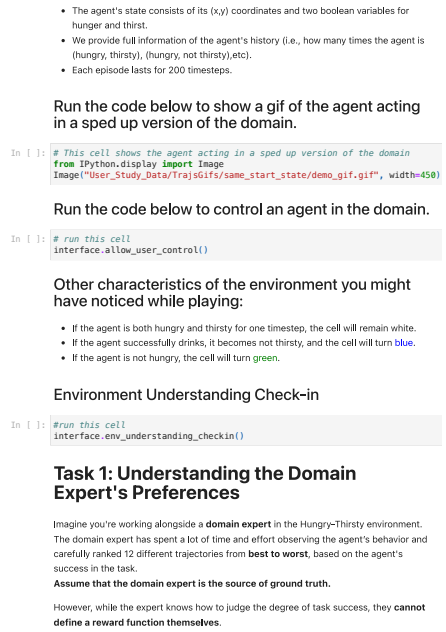
- If the agent drinks, it becomes not thirsty.
- At each timestep, the agent has a 10% probability of becoming thirsty.
- If the agent eats while not thirsty, it successfully eats the food and becomes not hungry for one timestep.
- The agent's goal is to be not hungry for as many timesteps as possible.

How Actions Can Fail:

- The drink action fails if the agent is not at the water location.
- The eat action fails if the agent is thirsty, or if the agent is not at the food location.
- The move action fails if the agent tries to move through one of the red barriers (depicted below).

Other Information:

(a) First page.



- The agent's state consists of its (x,y) coordinates and two boolean variables for hunger and thirst.
- We provide full information of the agent's history (i.e., how many times the agent is (hungry, thirsty), (hungry, not thirsty),etc).
- Each episode lasts for 200 timesteps.

Run the code below to show a gif of the agent acting in a sped up version of the domain.

In [ ]:
```
# This cell shows the agent acting in a sped up version of the domain
from IPython.display import Image
Image("User_Study_Data/TrajsGifs/same_start_state/demo_gif.gif", width=450)
```

Run the code below to control an agent in the domain.

In [ ]:
```
# run this cell
interface.allow_user_control()
```

Other characteristics of the environment you might have noticed while playing:

- If the agent is both hungry and thirsty for one timestep, the cell will remain white.
- If the agent successfully drinks, it becomes not thirsty, and the cell will turn blue.
- If the agent is not hungry, the cell will turn green.

Environment Understanding Check-in

In [ ]:
```
#run this cell
interface.env_understanding_checkin()
```

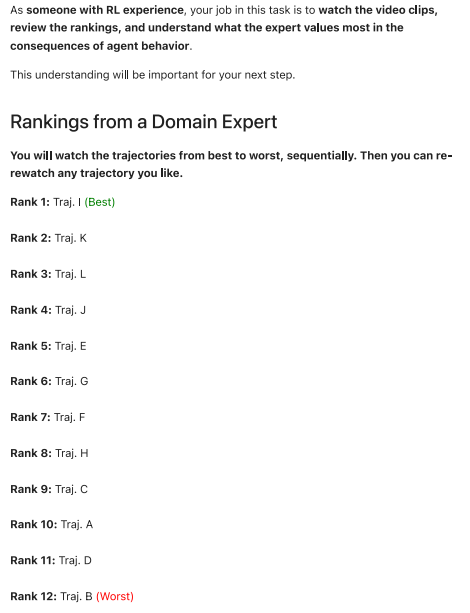
Task 1: Understanding the Domain Expert's Preferences

Imagine you're working alongside a **domain expert** in the Hungry–Thirsty environment. The domain expert has spent a lot of time and effort observing the agent's behavior and carefully ranked 12 different trajectories from **best to worst**, based on the agent's success in the task.
Assume that the domain expert is the source of ground truth.

However, while the expert knows how to judge the degree of task success, they **cannot define a reward function themselves**.

(b) Second page.

Figure 6: First and second pages of the UI in the human-subject study.



As **someone with RL experience**, your job in this task is to **watch the video clips, review the rankings, and understand what the expert values most in the consequences of agent behavior**.

This understanding will be important for your next step.

Rankings from a Domain Expert

You will watch the trajectories from best to worst, sequentially. Then you can re-rewatch any trajectory you like.

Rank 1: Traj. I (Best)

Rank 2: Traj. K

Rank 3: Traj. L

Rank 4: Traj. J

Rank 5: Traj. E

Rank 6: Traj. G

Rank 7: Traj. F

Rank 8: Traj. H

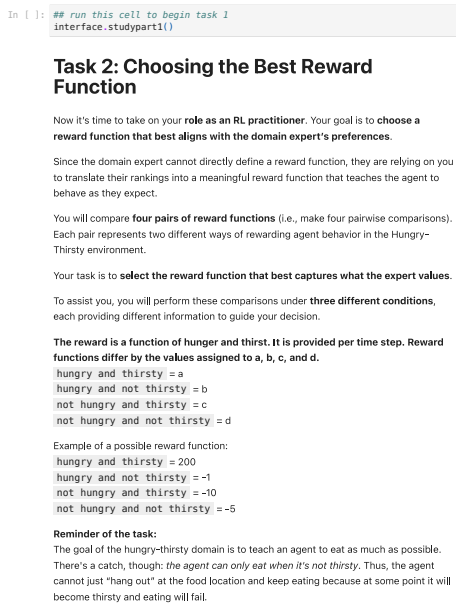
Rank 9: Traj. C

Rank 10: Traj. A

Rank 11: Traj. D

Rank 12: Traj. B (Worst)

(a) Third page.



In [ ]:
```
## run this cell to begin task 1
interface.studypart1()
```

Task 2: Choosing the Best Reward Function

Now it's time to take on your **role as an RL practitioner**. Your goal is to **choose a reward function that best aligns with the domain expert's preferences**.

Since the domain expert cannot directly define a reward function, they are relying on you to translate their rankings into a meaningful reward function that teaches the agent to behave as they expect.

You will compare **four pairs of reward functions** (i.e., make four pairwise comparisons). Each pair represents two different ways of rewarding agent behavior in the Hungry–Thirsty environment.

Your task is to **select the reward function that best captures what the expert values**.

To assist you, you will perform these comparisons under **three different conditions**, each providing different information to guide your decision.

The reward is a function of hunger and thirst. It is provided per time step. Reward functions differ by the values assigned to a, b, c, and d.
`hungry and thirsty` = a
`hungry and not thirsty` = b
`not hungry and thirsty` = c
`not hungry and not thirsty` = d

Example of a possible reward function:
`hungry and thirsty` = 200
`hungry and not thirsty` = −1
`not hungry and thirsty` = −10
`not hungry and not thirsty` = −5

Reminder of the task:
The goal of the hungry–thirsty domain is to teach an agent to eat as much as possible. There's a catch, though: *the agent can only eat when it's not thirsty*. Thus, the agent cannot just "hang out" at the food location and keep eating because at some point it will become thirsty and eating will fail.

(b) Fourth page.

Figure 7: Third and fourth pages of the UI in the human-subject study.

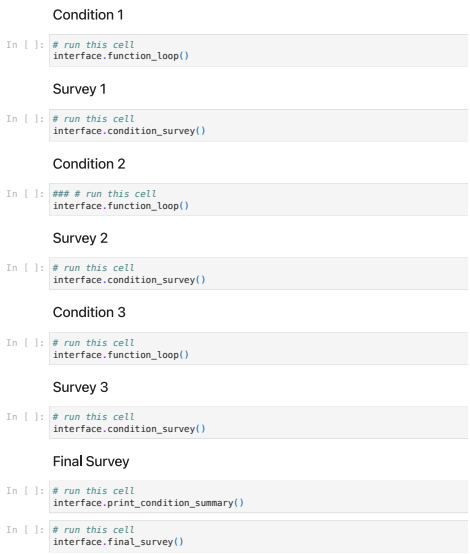


Figure 8: Fifth page of the UI in the human-subject study.

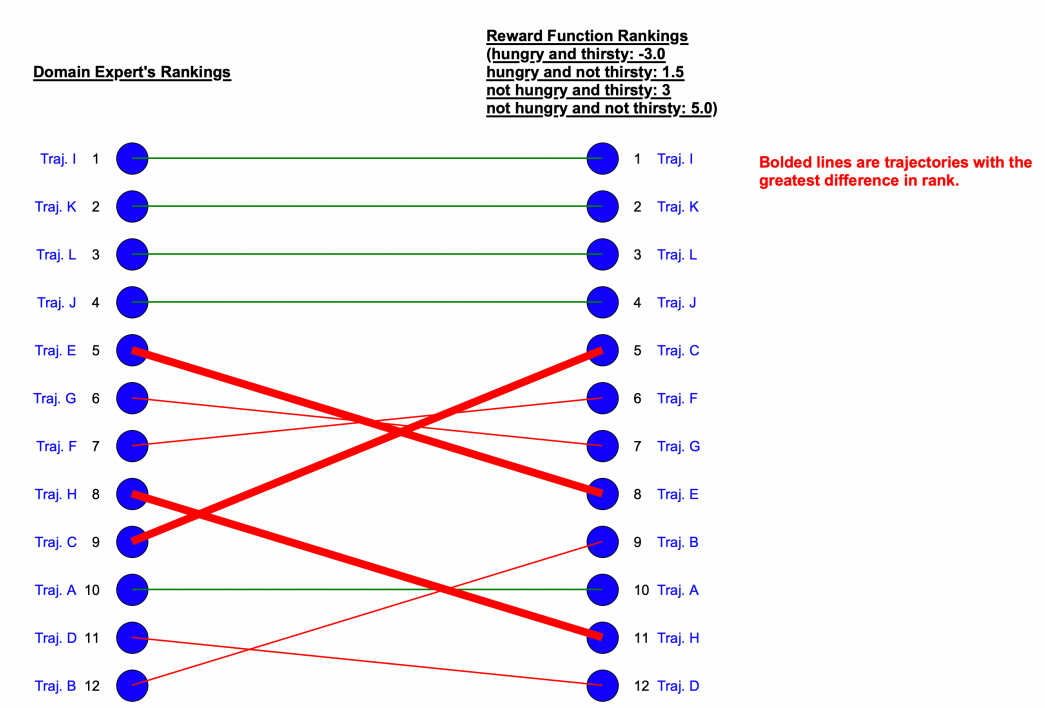
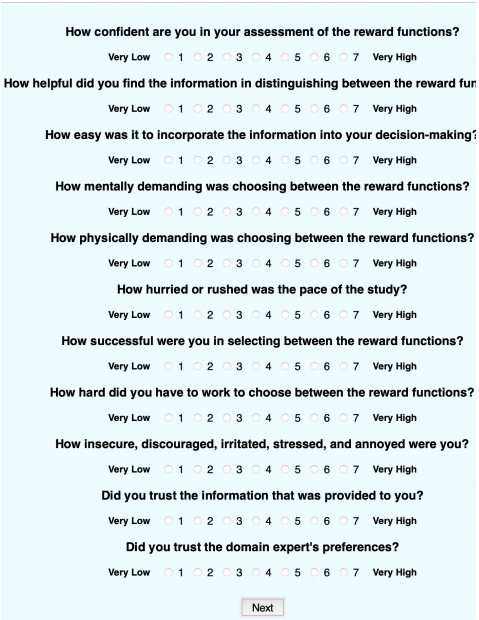
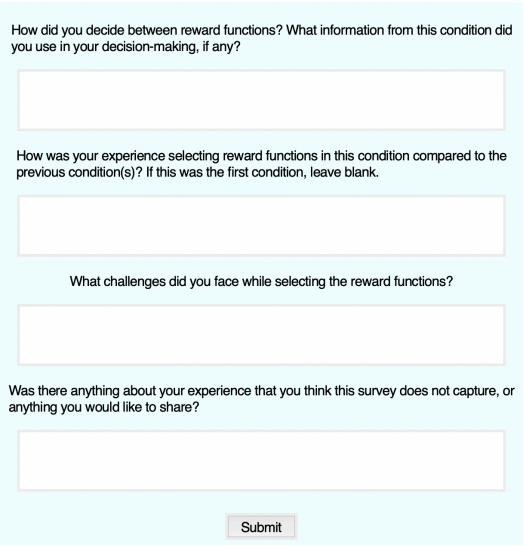


Figure 9: The visualization used in the Reward + Alignment + Visualization feedback condition. This shows how the preferences over trajectories differ between the domain expert and the reward function.

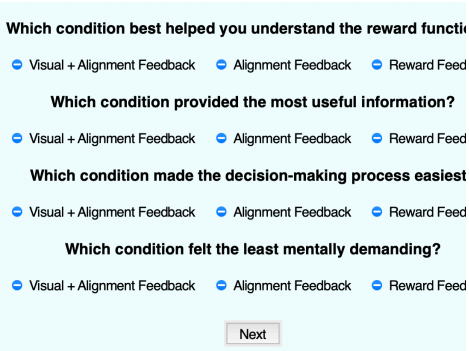


(a) Modified NASA TLX survey given to participants after each condition.

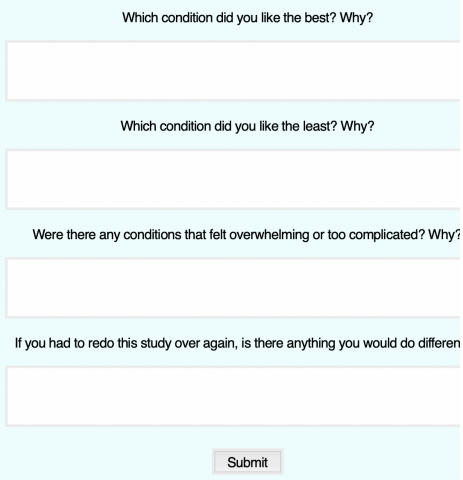


(b) Short answer survey given to participants after each condition.

Figure 10: Condition Experience Surveys.



(a) Final multiple choice survey given to participants to compare their experiences across the conditions.



(b) Final short answer survey given to participants to compare their experiences across the conditions.

Figure 11: Condition Comparison Surveys.

Next, in Tables 4 and 5, we present the reward functions used in the human-subject study. In the study, the reward functions in the REWARD FUNCTION 1 column were compared with the corresponding reward functions in the REWARD FUNCTION 2 column. We report the mean final return and the area under the curve (AUC) (both calculated from the evaluation metric), along with the standard deviation (STD). Note that the reward functions in both tables are identical; the only distinction is the metric being displayed (e.g., return or AUC).

REWARD FUNCTION 1	FINAL RETURN	REWARD FUNCTION 2	FINAL RETURN
(-0.9, -0.7, -0.4, 1.1)	39.138 ± 38.619	(-1, 0, 0.5, 1)	30.832 ± 43.009
(-3.7, 0.0, -3.1, 5.1)	33.495 ± 43.280	(-3.0, 1.5, 3, 5.0)	1.079 ± 0.319
(-0.9, -0.7, -0.4, 1.1)	39.138 ± 38.619	(-0.05, 0.2, 1.0, 1.0)	1.608 ± 7.617
(-3.6, 0.0, -3.1, 5.4)	35.485 ± 43.423	(-5.8, 1.2, 3.6, 5.8)	1.555 ± 4.537
(0, 0, 10, 10)	49.709 ± 34.312	(-0.05, 0.2, 1.0, 1.0)	1.608 ± 7.617
(-5.0, 0, 3.25, 5.0)	31.582 ± 43.205	(-5.0, 1.5, 3.25, 5.0)	1.225 ± 0.560
(-0.5, -0.5, 10.0, 10.0)	51.755 ± 36.960	(-0.05, 0.2, 1.0, 1.0)	1.608 ± 7.617
(-0.4, -0.5, 0.0, 1.0)	44.338 ± 41.750	(-0.2, 0.2, 0.5, 1.0)	1.217 ± 4.485
(-5.0, 0.0, -2.5, 5.0)	29.738 ± 42.468	(-5.0, 1.5, 3.25, 5.0)	1.225 ± 0.560
(-1.0, -0.05, -0.25, 1.0)	37.488 ± 44.081	(-5.0, 1.5, 3.25, 5.0)	1.225 ± 0.560
(-3.75, 0.0, -3.0, 5.0)	33.127 ± 43.348	(-5.0, 1.5, 3.25, 5.0)	1.225 ± 0.560
(-1.0, -0.7, -0.5, 1.0)	35.772 ± 36.955	(-0.05, 0.2, 1.0, 1.0)	1.608 ± 7.617

Table 4: This table shows the reward functions being compared in the reward selection aspect of the user study. We also report the mean final return ± STD.

REWARD FUNCTION 1	AUC	REWARD FUNCTION 2	AUC
(-0.9, -0.7, -0.4, 1.1)	240041.894 ± 278901.770	(-1, 0, 0.5, 1)	203405.861 ± 293167.199
(-3.7, 0.0, -3.1, 5.1)	219656.426 ± 296860.095	(-3.0, 1.5, 3, 5.0)	10619.689 ± 1872.266
(-0.9, -0.7, -0.4, 1.1)	240041.894 ± 278901.770	(-0.05, 0.2, 1.0, 1.0)	11390.661 ± 17391.171
(-3.6, 0.0, -3.1, 5.4)	227802.180 ± 296571.891	(-5.8, 1.2, 3.6, 5.8)	13451.407 ± 15346.038
(0, 0, 10, 10)	320899.367 ± 251997.175	(-0.05, 0.2, 1.0, 1.0)	11390.661 ± 17391.171
(-5.0, 0, 3.25, 5.0)	206036.191 ± 295113.945	(-5.0, 1.5, 3.25, 5.0)	11585.087 ± 3764.687
(-0.5, -0.5, 10.0, 10.0)	349167.541 ± 281517.767	(-0.05, 0.2, 1.0, 1.0)	11390.661 ± 17391.171
(-0.4, -0.5, 0.0, 1.0)	284235.135 ± 304976.538	(-0.2, 0.2, 0.5, 1.0)	10914.361 ± 15017.765
(-5.0, 0.0, -2.5, 5.0)	192430.552 ± 284398.848	(-5.0, 1.5, 3.25, 5.0)	11585.087 ± 3764.687
(-1.0, -0.05, -0.25, 1.0)	254052.926 ± 304591.173	(-5.0, 1.5, 3.25, 5.0)	11585.087 ± 3764.687
(-3.75, 0.0, -3.0, 5.0)	216262.144 ± 295986.126	(-5.0, 1.5, 3.25, 5.0)	11585.087 ± 3764.687
(-1.0, -0.7, -0.5, 1.0)	217347.509 ± 256831.683	(-0.05, 0.2, 1.0, 1.0)	11390.661 ± 17391.171

Table 5: This table shows the reward functions being compared in the reward selection aspect of the user study. We also report the mean AUC ± STD.

In an earlier version of the user study, two participants had a slightly different UI design. The differences included some variations in the wording of the instructions. We also did not include a game-play session that allowed user control in the domain. Additionally, we did not ask participants whether they trusted the domain expert or the information being provided to them. There were also minor changes in the reward functions considered. Initially, we had a set of 13 pairs of reward functions, from which we sampled 12 per participant. However, for simplicity in data analysis, we decided to remove one pair. We also replaced one reward function pair with another, to make the reward functions being compared more distinct.

C Proofs

Lemma C.1. *Given the infinite-horizon setting, if the expected returns under reward function r' are a positive linear transformation of the expected returns under reward function r, with respect to all*

trajectory distributions, then the preference ordering over any two trajectory distributions η_i and η_j remains unchanged. Formally:

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \alpha\mathbb{E}_{\tau\sim\eta}[G_r(\tau)] + \beta \implies \Big(\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \iff \eta_i \underset{(r',\gamma)}{\succsim} \eta_j\Big) \quad \forall \eta_i, \eta_j,$$

where $\alpha > 0$ and β are constants and the expectations $\mathbb{E}_{\tau\sim\eta}[G_r(\tau)]$ and $\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)]$ are taken over the same trajectory distributions.

Proof. Let η_i, η_j be arbitrary trajectory distributions. Without loss of generality, assume that $\eta_i \underset{(r,\gamma)}{\succsim} \eta_j$ From Definition 3, this implies:

$$\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \iff \mathbb{E}_{\tau\sim\eta_i}[G_r(\tau)] \geq \mathbb{E}_{\tau\sim\eta_j}[G_r(\tau)]$$

Define the difference in expected returns under r as:

$$\Delta_{i,j}\overline{G}(r) \doteq \mathbb{E}_{\tau\sim\eta_i}[G_r(\tau)] - \mathbb{E}_{\tau\sim\eta_j}[G_r(\tau)] \tag{C.1}$$

Now, consider the transformation of the expected return under r':

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \alpha\mathbb{E}_{\tau\sim\eta}[G_r(\tau)] + \beta,$$

where $\alpha > 0$ and β are constants. Define the corresponding difference under r':

$$\Delta_{i,j}\overline{G}(r') = \mathbb{E}_{\tau\sim\eta_i}[G_{r'}(\tau)] - \mathbb{E}_{\tau\sim\eta_j}[G_{r'}(\tau)] \tag{C.2}$$

Substitute the expressions for $\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)]$, we get:

$$\Delta_{i,j}\overline{G}(r') = \Big(\alpha\mathbb{E}_{\tau\sim\eta_i}[G_r(\tau)] + \beta\Big) - \Big(\alpha\mathbb{E}_{\tau\sim\eta_j}[G_r(\tau)] + \beta\Big)$$

Simplify the terms and notice that β cancels out. We obtain:

$$\Delta_{i,j}\overline{G}(r') = \alpha\Delta_{i,j}\overline{G}(r)$$

Now, consider the two cases:

Case 1: $\eta_i \underset{(r,\gamma)}{\succ} \eta_j$

This means $\Delta_{i,j}\overline{G}(r) > 0$. Since $\alpha > 0$, we have:

$$\Delta_{i,j}\overline{G}(r') = \alpha\Delta_{i,j}\overline{G}(r) > 0$$

As α is a positive constant, we conclude: $\eta_i \underset{(r',\gamma)}{\succ} \eta_j$.

Case 2: $\eta_i \underset{(r,\gamma)}{\sim} \eta_j$

This means $\Delta_{i,j}\overline{G}(r) = 0$. Apply the transformation to obtain:

$$\Delta_{i,j}\overline{G}(r') = \alpha \cdot 0 = 0$$

Thus, $\eta_i \underset{(r',\gamma)}{\sim} \eta_j$

Since both cases preserve the preference ordering, we conclude:

$$\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \iff \eta_i \underset{(r',\gamma)}{\succsim} \eta_j$$

□

Lemma C.2. *[Necessity] In the infinite horizon setting, if two trajectory distributions $\eta_i \in H(\mu_i)$ and $\eta_j \in H(\mu_j)$ have different start-state distributions ($\mu_i \neq \mu_j$), then there exists a potential function Φ such that:*

$$\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \text{ and } \eta_i \underset{(r',\gamma)}{\prec} \eta_j.$$

Proof. Let $\eta_i \in H(\mu_i)$ and $\eta_j \in H(\mu_j)$ be arbitrary trajectory distributions that have different start-state distributions ($\mu_i \neq \mu_j$). Without loss of generality assume that $\eta_i \underset{(r,\gamma)}{\succsim} \eta_j$. From Definition (3), this implies that:

$$\mathbb{E}_{\tau\sim\eta_i}[G_r(\tau)] \geq \mathbb{E}_{\tau\sim\eta_j}[G_r(\tau)]$$

We now analyze how the expected return changes under the potential-shaped reward function r'.

From Equation (4.4) we have

$$\mathbb{E}_{\tau\sim\eta_i(\tau)}[G_{r'}] = \mathbb{E}_{\tau\sim\eta_i(\tau)}[G_r(\tau)] - \mathbb{E}_{s_0\sim\mu_i}\left[\Phi(s_0)\right], \tag{C.3}$$

$$\mathbb{E}_{\tau\sim\eta_j(\tau)}[G_{r'}] = \mathbb{E}_{\tau\sim\eta_j(\tau)}[G_r(\tau)] - \mathbb{E}_{s_0\sim\mu_j}\left[\Phi(s_0)\right]. \tag{C.4}$$

Next, we define $\Delta_{i,j}\overline{G}(r), \Delta_{i,j}\overline{G}(r')$:

$$\Delta_{i,j}\overline{G}(r) \doteq \mathbb{E}_{\tau\sim\eta_i}[G_r(\tau)] - \mathbb{E}_{\tau\sim\eta_j}[G_r(\tau)] \tag{C.5}$$

$$\Delta_{i,j}\overline{G}(r) \doteq \mathbb{E}_{\tau\sim\eta_i}[G_{r'}(\tau)] - \mathbb{E}_{\tau\sim\eta_j}[G_{r'}(\tau)] \tag{C.6}$$

Substitute Equations (C.3) and (C.4) into Equation (C.6) to obtain $\Delta_{i,j}\overline{G}(r')$:

$$\Delta_{i,j}\overline{G}(r') = \Big(\mathbb{E}_{\tau\sim\eta_i(\tau)}[G_r(\tau)] - \mathbb{E}_{s_0\sim\mu_i}\left[\Phi(s_0)\right]\Big) - \Big(\mathbb{E}_{\tau\sim\eta_j(\tau)}[G_r(\tau)] - \mathbb{E}_{s_0\sim\mu_j}\left[\Phi(s_0)\right]\Big)$$

Rearrange terms and substitute in the equation for $\Delta_{i,j}\overline{G}(r)$, Equation (C.5):

$$\begin{aligned}\Delta_{i,j}\overline{G}(r') &= \Big(\mathbb{E}_{\tau\sim\eta_i(\tau)}[G_r(\tau)] - \mathbb{E}_{\tau\sim\eta_j(\tau)}[G_r(\tau)]\Big) - \Big(\mathbb{E}_{s_0\sim\mu_i}\left[\Phi(s_0)\right] - \mathbb{E}_{s_0\sim\mu_j}\left[\Phi(s_0)\right]\Big) \\ &= \Delta_{i,j}\overline{G}(r) - \Big(\mathbb{E}_{s_0\sim\mu_i}\left[\Phi(s_0)\right] - \mathbb{E}_{s_0\sim\mu_j}\left[\Phi(s_0)\right]\Big)\end{aligned} \tag{C.7}$$

Now we show that there exists a potential-based shaping function that will invert the preference ordering over trajectory distributions η_i, η_j, that is $\exists\Phi : \mathcal{S} \to \mathbb{R}$ such that $\eta_i \underset{(r',\gamma)}{\prec} \eta_j$.

From Definition 3, it follows that:

$$\eta_i \underset{(r',\gamma)}{\prec} \eta_j \iff \Delta_{i,j}\overline{G}(r') < 0$$

This provides the necessary condition for the existence of such a shaping function. Now let $\Delta_{i,j}\overline{\Phi}$ be defined as:

$$\Delta_{i,j}\overline{\Phi} \doteq \mathbb{E}_{s_0\sim\mu_i}\left[\Phi(s_0)\right] - \mathbb{E}_{s_0\sim\mu_j}\left[\Phi(s_0)\right] \tag{C.8}$$

Combine Equations (C.7) and (C.8) to get:

$$\Delta_{i,j}\overline{G}(r') = \Delta_{i,j}\overline{G}(r) - \Delta_{i,j}\overline{\Phi}$$

where $\Delta_{i,j}\overline{G}(r) > 0$, since $\eta_i \underset{(r,\gamma)}{\succsim} \eta_j$. Now it is clear that:

$$\Delta_{i,j}\overline{G}(r') < 0 \iff \Delta_{i,j}\overline{\Phi} > \Delta_{i,j}\overline{G}(r)$$

We now provide an example of a potential-based shaping function for which $\Delta_{i,j}\overline{\Phi} > \Delta_{i,j}\overline{G}(r)$. To begin, let us partition the state space $\mathcal{S}$ into two subsets: (1) the set of states that are more probable under μ_i and (2) those that are more probable under μ_j, denoted as $\mathcal{S}_{\mu_i>\mu_j}$ and $\mathcal{S}_{\mu_i\leq\mu_j}$, respectively:

$$\mathcal{S}_{\mu_i>\mu_j} = \{s|s \in \mathcal{S}, \mu_i(s) > \mu_j(s)\} \tag{C.9}$$
$$\mathcal{S}_{\mu_i\leq\mu_j} = \{s|s \in \mathcal{S}, \mu_i(s) \leq \mu_j(s)\} \tag{C.10}$$

Next, we define the potential-based shaping function $\Phi : \mathcal{S} \to \mathbb{R}$ as a piecewise function, which takes the following form:

$$\Phi(s) = \begin{cases} \frac{\Delta_{i,j}\overline{G}(r)+\epsilon}{\int_{s\in\mathcal{S}_{\mu_i>\mu_j}} \mu_i(s)-\mu_j(s)}, & \text{if } s \in \mathcal{S}_{\mu_i>\mu_j} \\ 0 & \text{if } s \in \mathcal{S}_{\mu_i\leq\mu_j} \end{cases} \tag{C.11}$$

where $\epsilon \in \mathbb{R}, \epsilon > 0$. For this shaping function, we define the difference in expected values as:

$$\Delta_{i,j}\overline{\Phi} \doteq \mathbb{E}_{s_0\sim\mu_i}\left[\Phi(s_0)\right] - \mathbb{E}_{s_0\sim\mu_j}\left[\Phi(s_0)\right]$$

We can express the expectation in integral form and rearrange the terms:

$$\begin{aligned} &= \int_{s\in\mathcal{S}} \mu_i(s)\Phi(s) - \int_{s\in\mathcal{S}} \mu_j(s)\Phi(s) \\ &= \int_{s\in\mathcal{S}} \big(\mu_i(s) - \mu_j(s)\big)\Phi(s) \end{aligned}$$

Decompose the integral over the two partitions defined in Equations (C.9) and (C.10). Notice that $\int_{s\in\mathcal{S}_{\mu_i\leq\mu_j}} \big(\mu_i(s) - \mu_j(s)\big)\Phi(s)$ goes to 0 by Equation (C.11):

$$\begin{aligned} &= \int_{s\in\mathcal{S}_{\mu_i>\mu_j}} \big(\mu_i(s) - \mu_j(s)\big)\Phi(s) + \int_{s\in\mathcal{S}_{\mu_i\leq\mu_j}} \big(\mu_i(s) - \mu_j(s)\big)\Phi(s) \\ &= \int_{s\in\mathcal{S}_{\mu_i>\mu_j}} \big(\mu_i(s) - \mu_j(s)\big)\Phi(s) \end{aligned}$$

Move $\Phi(s)$ outside of the integral as it is a constant by Equation (C.11) and simplify:

$$\begin{aligned} &= \frac{\Delta_{i,j}\overline{G}(r) + \epsilon}{\int_{s\in\mathcal{S}_{\mu_i>\mu_j}} \mu_i(s) - \mu_j(s)} \int_{s\in\mathcal{S}_{\mu_i>\mu_j}} \mu_i(s) - \mu_j(s) \\ &= \Delta_{i,j}\overline{G}(r) + \epsilon \end{aligned}$$

Hence $\Delta_{i,j}\overline{\Phi} = \Delta_{i,j}\overline{G}(r) + \epsilon > \Delta_{i,j}\overline{G}(r)$. □

Theorem C.3. *Given the infinite-horizon setting, the Trajectory Alignment Coefficient is invariant to positive linear transformations.*

Proof. Let $\{\eta_i, \eta_j\} \in \upsilon_h(D_h)$ be an arbitrary pair of trajectory distributions compared in the human preference dataset. Without loss of generality assume that $\eta_i^r \succsim \eta_j^r$. From Defintion (3), this implies that

$$\mathbb{E}_{\tau\sim\eta_i}[G_r(\tau)] \geq \mathbb{E}_{\tau\sim\eta_j}[G_r(\tau)].$$

We now analyze how the expected return changes under the reward function r'. The expected return under the reward function r is:

$$\mathbb{E}_{\tau\sim\eta}[G_r(\tau)] = \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})\right] \tag{C.12}$$

and the expected return under the shaped reward function r' is:

$$\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] = \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t r'(s_t, a_t, s_{t+1})\right] \tag{C.13}$$

Substitute $r'(s, a, s') = \alpha \cdot r(s, a, s') + \beta$ into Equation (C.13), we obtain:

$$\begin{aligned}\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] &= \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t(\alpha \cdot r(s_t, a_t, s_{t+1}) + \beta)\right] \\ &= \alpha \cdot \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t, s_{t+1})\right] + \mathbb{E}_{\tau\sim\eta}\left[\sum_{t=0}^{\infty}\gamma^t\beta\right]\end{aligned}$$

Since β is a constant, the expectation simplifies as follows:

$$\begin{aligned}\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)] &= \alpha \cdot \mathbb{E}_{\tau\sim\eta}[G_r(\tau)] + \beta\sum_{t=0}^{\infty}\gamma^t \\ &= \alpha \cdot \mathbb{E}_{\tau\sim\eta}[G_r(\tau)] + \frac{\beta}{1-\gamma}\end{aligned}$$

As $\mathbb{E}_{\tau\sim\eta}[G_{r'}(\tau)]$ is positive linear transformation of $\mathbb{E}_{\tau\sim\eta}[G_r(\tau)]$, we apply Lemma (C.1) to get:

$$\left(\eta_i \underset{(r,\gamma)}{\succsim} \eta_j \iff \eta_i \underset{(r',\gamma)}{\succsim} \eta_j\right)$$

Thus, from Equation (4.1) and Definition (5), we conclude that:

$$\sigma_{\text{TAC}}(D_h, D_{r,\gamma}) = \sigma_{\text{TAC}}(D_h, D_{r',\gamma})$$

□

D Environment Details

We use a modified Hungry-Thirsty domain (Singh et al., 2009), see Figure 12. The agent's start state is randomly placed at the beginning of each episode, while the food and water locations are randomly assigned per environment configuration (e.g., per run/seed). Lastly, reward functions take the form:

$$\begin{aligned}r(\text{hungry, thirsty}) &= a \\ r(\text{hungry, not thirsty}) &= b \\ r(\text{not hungry, thirsty}) &= c \\ r(\text{not hungry, not thirsty}) &= d\end{aligned}$$

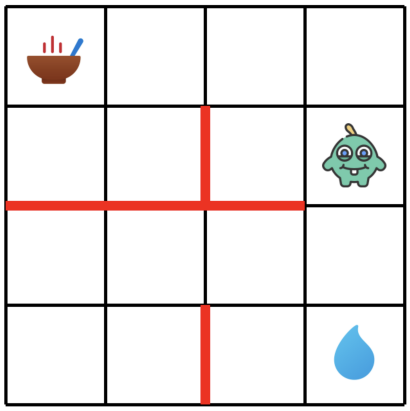


Figure 12: Hungry-Thirsty Environment