Analyzing Reward Functions via Trajectory Alignment

Calarina Muslimani
University of Alberta
musliman@ualberta.ca

Suyog Chandramouli
Princeton University
University of Alberta
suyoghc@princeton.edu

Serena Booth
Brown University
serena_booth@brown.edu

W. Bradley Knox
University of Texas at Austin
bradknox@cs.utexas.edu

Matthew E. Taylor
University of Alberta
Alberta Machine Intelligence Institute (Amii)
matthew.e.taylor@ualberta.ca

Abstract

Reward design in reinforcement learning (RL) is often overlooked, with the assumption that a well-defined reward is readily available. However, reward functions can be challenging to design and prone to reward hacking, potentially leading to unintended or dangerous consequences in real-world applications. To create safe RL agents, *reward alignment* is crucial. We define reward alignment as the process of designing reward functions that preserve the preferences of a human stakeholder. In practice, reward functions are designed with training performance as the primary measure of success; this measure, however, may not reflect alignment. This work studies the practical implications of reward design on alignment. Specifically, we (1) propose a reward alignment metric, the *Trajectory Alignment coefficient*, that measures the similarity between the preference orderings of a human stakeholder and the preference orderings induced by a reward function, (2) use this metric to quantify the prevalence and extent of misalignment in human-designed reward functions, and (3) examine how misalignment affects the efficacy of these human-designed reward functions in terms of training performance.

1 Introduction

In reinforcement learning (RL), the reward hypothesis states "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)" [1]. More generally, this means that RL can be used to solve a task given a reward function that properly specifies this task. However, the reward hypothesis does not address the practical challenges of designing fully specified reward functions. In practice, reward design is a difficult and error-prone process carried out by human engineers [2–4], in which reward hacking may be a common occurrence. Reward hacking occurs when an RL agent discovers and exploits unintended shortcuts in the reward function. CoastRunners is a prominent example, in which the designed reward encouraged RL agents to collect power boosts at the expense of crashing into other boats [5]. While CoastRunners is just a game, shortcuts can have dangerous consequences for real-world applications of RL, such as autonomous driving or water treatment.

To ensure safe and effective RL, reward alignment, wherein reward functions encode the preferences of a human stakeholder, is essential. However, a common approach to reward design involves an ad-hoc trial-and-error process, where RL practitioners iterate through different reward functions and evaluate their effectiveness based on the algorithm's training performance [3]. It has been reported that 92% of surveyed RL experts and 100% of the RL practitioners in an autonomous driving survey used this method [3, 6]. Additionally, large language models for reward design often rely solely

NeurIPS 2024 Workshop on Behavioral Machine Learning.

on training/testing accuracy as their evaluation metric [7]. Overall, reward alignment is not often explicitly considered in reward design. Therefore, given the prevalence of ad-hoc reward design methods, such as trial and error, it is important to understand their consequences for reward alignment.

To that end, we propose the *Trajectory Alignment coefficient*, a novel reward alignment metric based on the similarity of humans' preference orderings to those induced by a given reward function and discount factor. Unlike prior work [4, 8, 9], our metric is: (1) expressive, as it quantifies the degree of reward alignment rather than only detecting misalignment, and (2) capable of evaluating reward alignment solely based on human preferences, eliminating the need for a ground-truth reward function. We then use our proposed metric to analyze human-designed reward functions and find regular occurrences of misalignment. Overall, our goal is to highlight the prevalence of misalignment in human-designed reward functions and to encourage researchers to prioritize reward alignment in the reward design process.

2 Related Work

Early alignment research often focused on directly training agents to align with human preferences [10]. Recent efforts have added a focus on evaluating the quality of both engineered and learned reward functions. For example, one line of research has identified empirical investigations to understand the shortcomings of current reward design practices and evaluation schemes [3, 6, 11]. A second line in reward learning has developed metrics to quantify the differences between learned and ground truth reward functions [8, 9]. These metrics allow for a direct comparison of reward functions without the need for computationally expensive policy evaluations. However, a key limitation is that to assess reward alignment, these metrics require access to a ground truth reward function that accurately reflects the preferences of a human stakeholder. A third research direction has focused on verifying the alignment of an RL agent's behavior [12] where reward functions are defined as aligned if and only if they induce the same set of optimal policies. We argue that defining alignment only with respect to optimal policies can be limiting, particularly in online learning settings where one cares about the agent's lifetime performance. Our work closely relates to recent work on specifying human-aligned RL objectives [4]. This prior work developed approaches to identifying misalignment in RL objectives and outlined common causes of misalignment. Importantly, our work differs in that we propose a more expressive metric that captures the degree of reward function alignment rather than solely confirming its existence. Our metric offers a more nuanced assessment of the quality of the reward function, enabling more effective reward design. Lastly, while LLM alignment is also a prominent field, it is beyond the scope of this work due to its broad focus, which can include mitigating adversarial attacks, detecting bias, and ensuring interpretability.

3 Background

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 states and actions, with a return defined as the sum of discounted future rewards: $G(\tau) = \sum_{t=0}^{|\tau|-1} \gamma^t r_{t+1}$ with discount factor $\gamma \in [0, 1)$. The agent attempts to learn a policy, $\pi : \mathcal{S} \to \mathcal{A}$, to maximize the expected return.

3.1 Reward Functions Induce Preference Orderings

Let us first consider the deterministic case, by which preferences are based on individual trajectories. In this case, given a fixed reward function and discount factor, a preferred trajectory is one that yields a greater return:

$$\tau_{\text{A}} \succ \tau_{\text{B}} \iff G(\tau_{\text{A}}) > G(\tau_{\text{B}}) \tag{1}$$

We now shift to the stochastic setting, which considers probability distributions over trajectories[1]. This arises when the environment or the agent's behavior is stochastic. Let $\eta_{\text{A}}, \eta_{\text{B}} \in H$ be probability distributions over all possible trajectories. In this case, given a reward function and discount factor:

$$\eta_{\text{A}} \succ \eta_{\text{B}} \iff \mathbb{E}_{\tau_{\text{a}} \sim \eta_{\text{A}}}[G(\tau_{\text{a}})] > \mathbb{E}_{\tau_{\text{b}} \sim \eta_{\text{B}}}[G(\tau_{\text{b}})] \tag{2}$$

[1]Note that the stochastic case inherently subsumes the deterministic case.

Intuitively, equations 1 and 2 state that reward function and discount factor pairs naturally induce a preference ordering over trajectories (or trajectory distributions) based on the expected return. To illustrate these concepts, consider the simple autonomous driving task [4] in Figure 1. Suppose there exists only three trajectories $\{\tau_{\text{success}}, \tau_{\text{idle}}, \tau_{\text{crash}}\}$, and a trajectory distribution η. τ_{success} consists of safe driving. τ_{crash} consists of a car crashing, and τ_{idle} consists of a car remaining parked. $\eta_{\text{success-crash}}$ samples τ_{success} with a 90% probability, and τ_{crash} with a 10% probability. Next, consider a reward function and discount with return values: $G(\tau_{\text{success}}) = 10$, $G(\tau_{\text{idle}}) = 0$, $G(\tau_{\text{crash}}) = -50$, $G(\eta_{\text{success-crash}}) = 4$. Based on equations 1 and 2, the resulting preference ordering is $\tau_{\text{success}} \succ \eta_{\text{success-crash}} \succ \tau_{\text{idle}} \succ \tau_{\text{crash}}$.

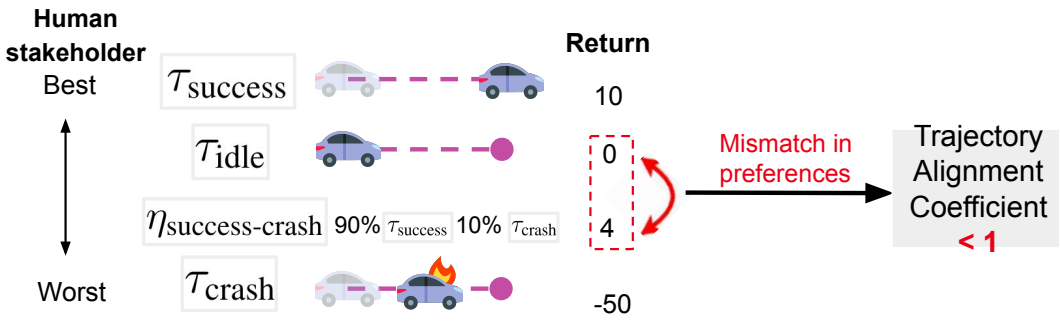


Figure 1: This figure provides an example of the trajectory alignment coefficient in the case of a simple autonomous driving scenario.

4 Characterizing Reward Alignment

This section first provides a formal definition for *Total Trajectory Alignment*, a concept rooted in prior work on policy preferences [13]. Next, we introduce a practical measure of total trajectory alignment.

Definition 1. *Given trajectory distribution $\eta \in H$ induced by an environment e and policy π, we say a reward function r is* total trajectory aligned *if the human stakeholder prefers*

$$\eta_i \succ \eta_j \quad \forall i, j \iff \mathbb{E}_{\tau\sim\eta_i}[G(\tau)] \succ \mathbb{E}_{\tau\sim\eta_j}[G(\tau)] \quad \forall i, j \tag{3}$$

Intuitively, definition 1 states that a reward function is total trajectory aligned if and only if it induces the same preference ordering over all possible trajectory distributions as the human stakeholder. We further note that this definition of total trajectory alignment can be extended to evaluating alignment between two reward functions. Specifically, two reward functions (and discounts) are total trajectory aligned if and only if the respective preference orderings over all possible trajectory distributions are identical.

Kendall's Tau to Measure Total Trajectory Alignment Definition 1 can determine whether a human-designed reward is total trajectory aligned. However, to effectively compare and improve reward functions, we need a quantitative metric that measures the degree of alignment between a reward function and a human stakeholder. To that end, we propose the *Trajectory Alignment coefficient*, a reward alignment measure based on Kendall's Tau correlation. Kendall's Tau is a non-parametric measure that quantifies the level of agreement between two sets of ranked data [14][2]. Its output is 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 then measures the similarity among preference orderings over all possible trajectory distributions. By definition 1, only a trajectory alignment coefficient of 1 signifies total trajectory alignment.

Trajectory Alignment Coefficient in Practice Total trajectory alignment and the trajectory alignment coefficient can, in theory, entail an intractably large ordering over trajectory distributions. Therefore, to practically apply the trajectory alignment coefficient as a reward alignment measure,

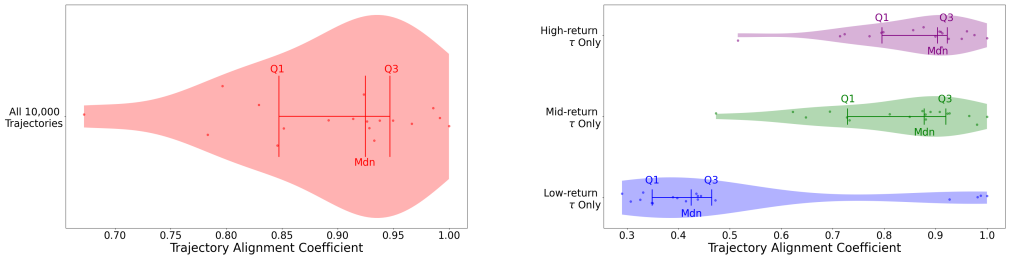
[2]We use Kendall's Tau-b variant.

we must consider tractable-sized subsets of these trajectories. This requires specifying the number of trajectories to rank and the trajectory sampling method. To demonstrate this concept, consider the autonomous driving task in Figure 1 again. The toy reward function, discount factor 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: $\tau_{\text{success}} \succ \tau_{\text{idle}} \succ \eta_{\text{success-crash}} \succ \tau_{crash}$. This results in a trajectory alignment coefficient < 1, highlighting some misalignment in the reward.

5 Analysis of Human-Designed Reward Functions

To analyze the alignment of human-designed reward functions, we consider the open-sourced reward data from a human subject study of 18 self-identified RL experts [3]. In this study, participants engaged in the reward design for a simple 4×4 grid-world called Hungry-Thirsty [15]. See Appendix A for environment details. To simplify the reward design process, participants selected weights for the four state features within a predetermined interval. Participants were instructed to select a reward function, algorithm, and hyperparameters to train an RL agent capable of solving the given task. A clearly defined evaluation metric (i.e., ground truth reward) was also provided to the participants that measured task success. After configuring the RL components, participants trained their agent and observed its performance based on both the evaluation metric and the chosen reward function. Ad-hoc trial-and-error reward design was a common practice amongst the participants, with each participant testing an average of 4.1 unique reward functions. Our analysis considers only the final reward function submitted (per participant).

Experiment Details For experimental purposes, we assume that the task evaluation metric represents the preferences of a fictitious human stakeholder. Note that this assumption simplifies our reward analysis and is not a fundamental limitation of our method. We evaluated the alignment between the reward functions designed by participants and this metric. To calculate the trajectory alignment coefficient, we select 75 trajectories and rank them using both the human-designed reward functions and the task evaluation metric. We select trajectories from low, medium, and high-performing policies (e.g., based on average return under the evaluation metric). See additional details on, and justification for, the trajectory selection procedure in Appendix B.2.



(a) (b)

Figure 2: In these figures, we analyze the extent of misalignment among 18 human-designed reward functions when considering different types of trajectories. Vertical lines indicate the 25th quartile (Q1), the median (Mdn), and the 75th percentile (Q3), respectively. The horizontal line indicates the interquartile range.

How Aligned are Human-Designed Reward Functions? To better understand the alignment across all human-designed reward functions, we measured the trajectory alignment coefficient using four different sets of trajectories: low-return only, medium-return only, high-return only, and all trajectories. The corresponding results are visualized in violin plots in Figure 2.

Across all our analyses, trajectory alignment coefficients ranged from 0.29 to 1.0, indicating a substantial degree of misalignment. Only 1 out of 18 participants successfully crafted a reward function that aligned with the evaluation metric. This means that in the other 94% of cases, the designed reward function led to preferences over the trajectories that differed from the preferences of the evaluation metric. These findings highlight that designing trajectory-aligned reward functions

is a significant challenge, even in simple grid worlds. This further emphasizes the need to consider reward alignment in the design pipeline, as trial-and-error approaches can be insufficient.

Moreover, in Figure 2b, we found that considering low-return trajectories (blue violin plot) produced significantly lower trajectory alignment coefficients than high-return trajectories (purple violin plot). These findings reveal two useful insights. Firstly, approaches focusing solely on optimal or high-return trajectories for evaluating reward alignment, such as Brown et al. [12], may not provide a complete picture of alignment. This is evident in the stark differences observed in the distribution of trajectory alignment scores when considering low-return and high-return trajectories separately. If high-return trajectories were sufficient for assessment, the distributions of scores should be similar. Secondly, prioritizing an understanding of how reward functions influence sub-optimal trajectories could be advantageous during the reward refinement process. The trajectory alignment coefficients were lowest for low-return trajectories, which indicates that the designed reward functions and evaluation metric had the greatest mismatch in preferences when considering these trajectories. This suggests that these reward functions may be incorrect in assessing sub-optimal state-action pairs. Adjusting these reward functions to better align sub-optimal trajectories with the evaluation metric could lead to significant gains in overall reward function alignment and, consequently, improve performance.

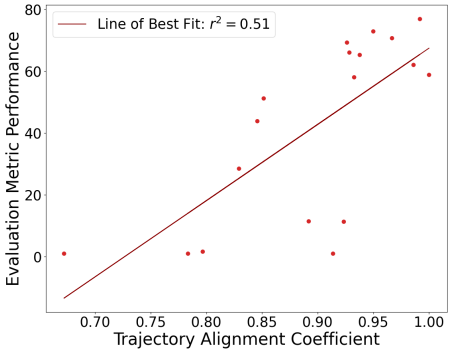


Figure 3: This figure highlights the relationship between the trajectory alignment coefficient and performance. We observe that less-aligned reward functions produce poor RL policies.

Is Reward Alignment Related to Policy Performance? RL seeks to learn high-performing policies. Therefore, it is critical to understand whether more aligned reward functions help achieve this goal. To do so, we trained one Q-Learning agent with each human-designed reward function for 10,000 episodes. We then performed rollouts using the final policies learned and evaluated the performance with the task evaluation metric. Figure 3 visualizes the trajectory alignment coefficient (using all 10,000 trajectories) versus the respective performance across all participant reward functions. Our findings indicate a moderate correlation between trajectory alignment coefficients and evaluation performance, with a coefficient of determination of 0.51. We also found that this relationship holds when considering the other trajectory subsets; see Appendix C.2. This suggests that RL practitioners prioritizing reward alignment are more likely to achieve high-performing policies.

6 Conclusion

To ensure the safe deployment of RL agents, it is important to design reward functions that align with human preferences. However, reward alignment is often overlooked in reward design in favor of other performance metrics. Therefore, this work investigates the consequences of practical reward design on alignment. By introducing a novel reward alignment measure, the trajectory alignment coefficient, we found that human-designed reward functions tend to be misaligned. We also found a correlation between reward alignment and performance, suggesting that prioritizing reward alignment can contribute to the effectiveness of RL agents. Future work should focus on incorporating reward alignment into the design pipeline and developing best-practice approaches to designing aligned rewards.

7 Acknowledgements

Part of this work has taken place in the Intelligent Robot Learning (IRL) Lab at the University of Alberta, which is supported in part by research grants from the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Digital Research Alliance of Canada; Mitacs; and NSERC. This work has also taken place in part in the Rewarding Lab at UT Austin. The Rewarding Lab is supported by NSF (IIS-2402650), ONR (N00014-22-1-2204), EA Ventures, Bosch, and UT Austin's Good Systems grand challenge.

References

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

[2] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*, volume 35, pages 9460–9471, 2022.

[3] 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 *Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023.

[4] W. Bradley Knox and James MacGlashan. How to specify reinforcement learning objectives. In *Reinforcement Learning Conference*, volume 1, 2024.

[5] Jack Clark and Dario Amodei. Faulty reward functions in the wild, 2016. URL `https://openai.com/research/faulty-reward-functions`.

[6] W. Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (mis)design for autonomous driving. *Artificial Intelligence*, 316:103829, 2023. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2022.103829. URL `https://www.sciencedirect.com/science/article/pii/S0004370222001692`.

[7] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=10uNUgI5Kl`.

[8] Blake Wulfe, Ashwin Balakrishna, Logan Ellis, Jean Mercat, Rowan McAllister, and Adrien Gaidon. Dynamics-aware comparison of learned reward functions. *arXiv preprint arXiv:2201.10081*, 2022.

[9] Adam Gleave, Michael Dennis, Shane Legg, Stuart Russell, and Jan Leike. Quantifying differences in reward functions. In *Tenth International Conference on Learning Representations*, 2021.

[10] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[11] Nils Wilde and Javier Alonso-Mora. Do we use the right measure? challenges in evaluating reward learning algorithms. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1553–1562. PMLR, 14–18 Dec 2023. URL `https://proceedings.mlr.press/v205/wilde23a.html`.

[12] Daniel S. Brown, Jordan Schneider, and Scott Niekum. Value alignment verification. In *International Conference on Machine Learning*, 2021. https://proceedings.mlr.press/v139/brown21a/brown21a.pdf.

[13] Michael Bowling, John D. Martin, David Abel, and Will Dabney. Settling the reward hypothesis, 2023. https://arxiv.org/pdf/2212.10420.

[14] Maurice George Kendall. *Rank correlation methods.* Griffin, 1948.

[15] Satinder Singh, Richard L Lewis, and Andrew G Barto. Where do rewards come from? In *Proceedings of the Annual Conference of the Cognitive Science Society*, pages 2601–2606. Cognitive Science Society, 2009.

Appendix

A Environment Details

We use a modified Hungry-Thirsty domain [15]. It is a 4×4 grid-world with a fixed time horizon of 200 steps. In this environment, there is food and water placed at random corners of the grid. We note that the food and water locations changes per seed/run, not per episode. The state space consists of four dimensions: the agent's x and y coordinates, and two Boolean variables representing hunger and thirst. The action space is one dimensional which includes the following options: moving up, down, left or right, eat, or drink. The goal is to maximize the time spent without experiencing hunger. The agent is hungry if and only if it did not eat in the previous time step. However, the agent can only eat if it is located at a food source and is not thirsty. Otherwise, if the agent chooses the eat action, then the agent will remain hungry. In addition, on every time-step the agent will randomly become thirsty with probability$= 0.10$. The evaluation metric for this task is based on the number of time-steps the agent is not hungry: Eval$[\tau] = \sum_{i=1}^{200} \neg(\text{hungry})$. In the the open-sourced reward data [3], the reward functions take the form:

$$\begin{aligned} r(\text{hungry, thirsty}) &= a \\ r(\text{hungry, not thirsty}) &= b \\ r(\text{not hungry, thirsty}) &= c \\ r(\text{not hungry, not thirsty}) &= d \end{aligned} \tag{4}$$

Participants then selected values for $a, b, c, d \in [-1, 1]$ by increments of 0.05.

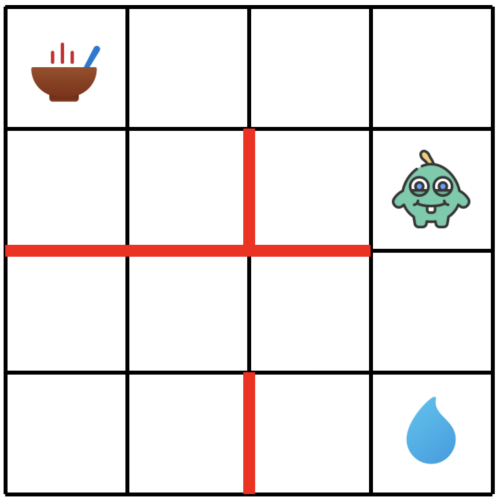


Figure 4: Hungry-Thirsty Environment

B Experiment Details

B.1 RL Training

An overview of the hyperparameters used for training the Q-Learning algorithm is provided in Table 1.

Table 1: Hyperparameters for Q-Learning algorithm.

Hyperparameter	Value
number of training episodes	10000
number of seeds	10
learning rate	0.05
exploration strategy	epsilon-greedy
epsilon	0.15
discount	0.99

B.2 Trajectory Sampling Protocol

To use our trajectory alignment coefficient as a reward alignment measure, two parameters must be specified: the number of trajectories to rank and the trajectory sampling method. For experiments in Section 5, we decided to select trajectories from three different policies, low-return, medium-return, and high-return. We used the following protocol to gather such trajectories and calculate the trajectory alignment coefficient:

1. Train a Q-Learning agent for 10000 episodes. After each episode, perform offline evaluation (i.e., policy roll out with no exploration). Store the corresponding trajectory and return under the evaluation metric.
2. Using the return values across all 10000 episodes, calculate the bottom 20th and top 80th percentile. See Table 2 for the respective scores.
3. Group the 10000 trajectory, return pairs by the return values. More specifically, if the return is less than the bottom 20th percentile, the trajectory is placed in the low-return group. If the return is greater than the top 80th percentile, the trajectory is placed in the high-return group. Otherwise, the trajectory is placed in the medium-return group.
4. Randomly sample 75 trajectories from each group.
5. For each group of trajectories, calculate the trajectory alignment coefficient.
6. Repeat steps 1-4 for each of the 10 seeds.
7. For each group, average the trajectory alignment coefficient across the 10 seeds. This result is what is shown in Figure 2 from Section 5.

We used this sampling strategy for two primary reasons. First, we wanted to understand how the trajectory set influences the trajectory alignment coefficient. Considering different sets of trajectories in the analysis provides possible insight on where more or less misalignment is occurring within a human-designed reward function. For example, we found more misalignment (with respect to our trajectory alignment coefficient) when using low-return trajectories as compared to higher-return trajectories. This might suggest that when refining the human-designed reward functions in the dataset [3], it is useful to prioritize understanding the affect of the reward functions on sub-optimal trajectories. Our second objective was to analyze the relationship between policy performance and alignment. Therefore, we aimed to determine whether our proposed metric consistently identifies such a relationship across different trajectory subsets.

C Additional Results

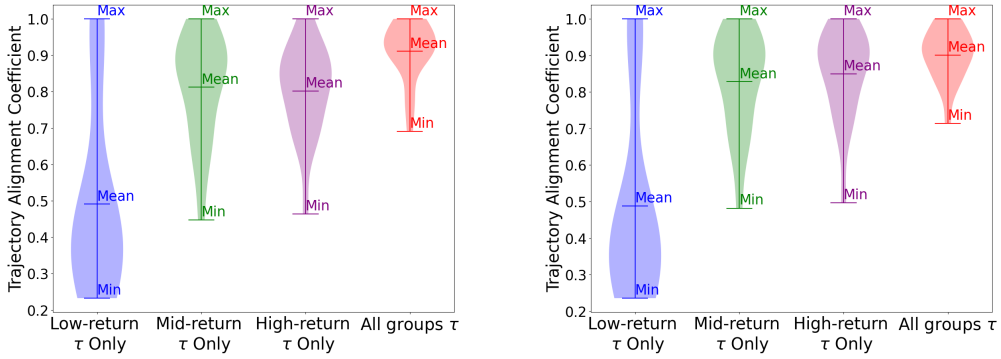
C.1 Effect of Trajectory Count on the Trajectory Alignment Coefficient

In this experiment, we measured the trajectory alignment coefficient using the four different sets of trajectories: low-return only, medium-return only, high-return only, and a combination of trajectories

Table 2: Return values corresponding to the bottom 20th and top 80th percentile.

Seed	20th Percentile	80th Percentile
0	59.0	89.0
1	1.0	52.0
2	1.0	53.0
3	4.0	54.0
4	57.0	89.0
5	4.0	37.0
6	12.80	87.0
7	23.0	65.0
8	59.0	89.0
9	6.0	37.0

sampled from all three groups. This is the same setup as in Section 5. However, this experiment ablates over the number of trajectories sampled. We choose from the set $\in [6, 300]$. In particular, we wanted to determine whether we observed similar findings as in Figure 2, when using a small number of trajectories. A smaller trajectory budget becomes increasingly important when we require a human stakeholder to perform the ranking. In Figure 5a, we found that even when we only sample 6 trajectories, we still observe similar trajectory coefficients as compared to sampling 300 trajectories in Figure 5b.



(a) Sampling Size=6

(b) Sampling Size=300

Figure 5: In these figures, we analyze the effect of trajectory sampling size on the observed trajectory alignment coefficient.

C.2 Effect of Trajectory Type on the Alignment-Performance Relationship

In this analysis, our goal is to understand whether the trajectory alignment coefficient is robust to the type of trajectories used in its calculation. More specifically, does the correlation between policy performance and the trajectory alignment coefficient hold when we consider different trajectory types. In Figures 6-8, we found that human-designed reward functions that achieved higher evaluation scores also had higher trajectory alignment coefficients.

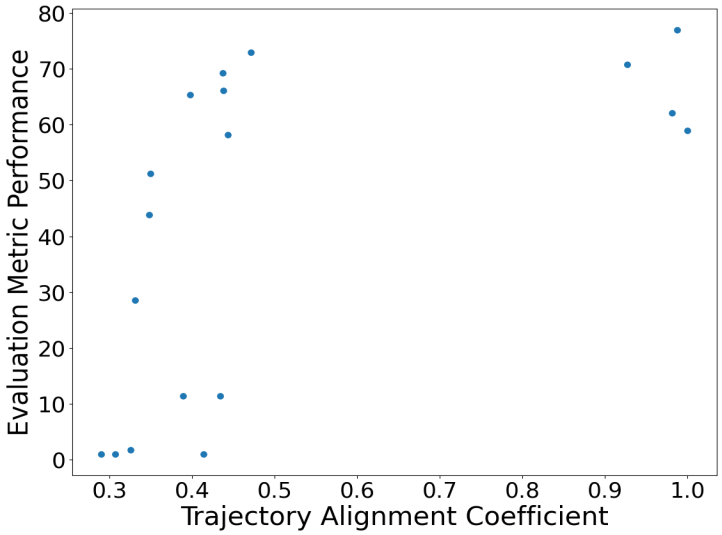


Figure 6: The x-axis is final performance of a Q-Learning trained with each human-designed reward function and evaluated with the ground truth reward. The y-axis is the trajectory alignment coefficient when only considering 75 *low-return trajectories.*

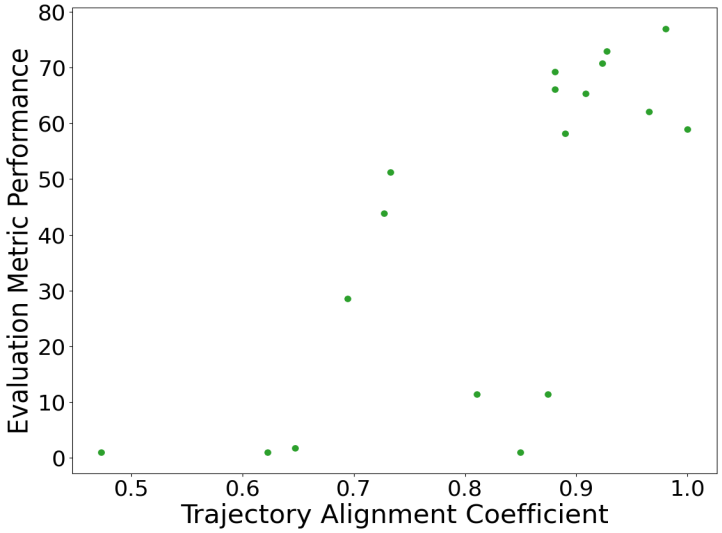


Figure 7: The x-axis is final performance of a Q-Learning trained with each human-designed reward function and evaluated with the ground truth reward. The y-axis is the trajectory alignment coefficient when only considering 75 *mid-return trajectories.*

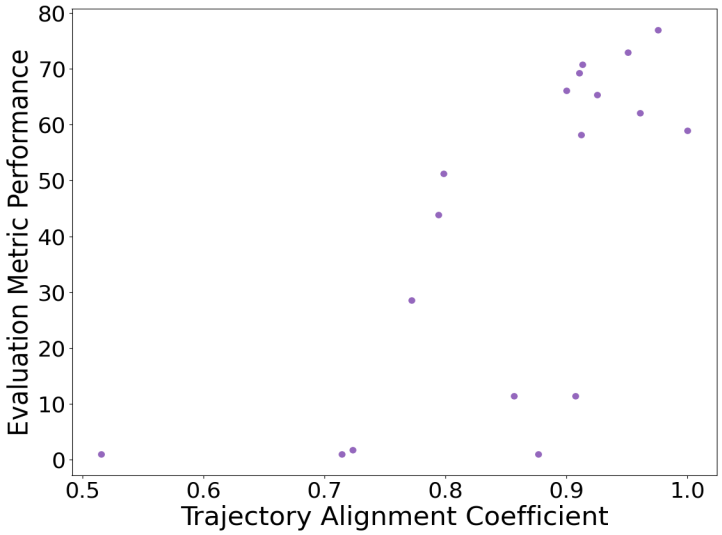


Figure 8: The x-axis is final performance of a Q-Learning trained with each human-designed reward function and evaluated with the ground truth reward. The y-axis is the trajectory alignment coefficient when only considering 75 *high-return trajectories*.