STARC: A GENERAL FRAMEWORK FOR QUANTIFYING DIFFERENCES BETWEEN REWARD FUNCTIONS

Joar Skalse
Department of Computer Science
Future of Humanity Institute
Oxford University
joar.skalse@cs.ox.ac.uk

Lucy Farnik
University of Bristol
Bristol AI Safety Centre
lucy.farnik@bristol.ac.uk

Sumeet Ramesh Motwani
Berkeley Artificial Intelligence Research
University of California, Berkeley
motwani@berkeley.edu

Erik Jenner
Berkeley Artificial Intelligence Research
University of California, Berkeley
jenner@berkeley.edu

Adam Gleave
FAR AI, Inc.
adam@far.ai

Alessandro Abate
Department of Computer Science
Oxford University
aabate@cs.ox.ac.uk

ABSTRACT

In order to solve a task using reinforcement learning, it is necessary to first formalise the goal of that task as a *reward function*. However, for many real-world tasks, it is very difficult to manually specify a reward function that never incentivises undesirable behaviour. As a result, it is increasingly popular to use *reward learning algorithms*, which attempt to *learn* a reward function from data. However, the theoretical foundations of reward learning are not yet well-developed. In particular, it is typically not known when a given reward learning algorithm with high probability will learn a reward function that is safe to optimise. This means that reward learning algorithms generally must be evaluated empirically, which is expensive, and that their failure modes are difficult to anticipate in advance. One of the roadblocks to deriving better theoretical guarantees is the lack of good methods for *quantifying* the difference between reward functions. In this paper we provide a solution to this problem, in the form of a class of pseudometrics on the space of all reward functions that we call STARC (STAndardised Reward Comparison) metrics. We show that STARC metrics induce both an upper and a lower bound on worst-case regret, which implies that our metrics are tight, and that any metric with the same properties must be bilipschitz equivalent to ours. Moreover, we also identify a number of issues with reward metrics proposed by earlier works. Finally, we evaluate our metrics empirically, to demonstrate their practical efficacy. STARC metrics can be used to make both theoretical and empirical analysis of reward learning algorithms both easier and more principled.

1 INTRODUCTION

To solve a sequential decision-making task with reinforcement learning or automated planning, we must first formalise that task using a reward function (Sutton & Barto, 2018; Russell & Norvig, 2020). However, for many tasks, it is extremely difficult to manually specify a reward function that captures the task in the intended way. To resolve this issue, it is increasingly popular to use *reward learning*, which attempts to *learn* a reward function from data. There are many techniques for doing this. For example, it is possible to use preferences between trajectories (e.g. Christiano et al., 2017), expert demonstrations (e.g. Ng & Russell, 2000), or a combination of the two (e.g. Ibarz et al., 2018).

To evaluate a reward learning method, we must *quantify* the *difference* between the learnt reward function and the underlying true reward function. However, doing this is far from straightforward. A simple method might be to measure their L_2-distance. However, this is unsatisfactory, because two reward functions can have a large L_2-distance, even if they induce the *same* ordering of policies, or a small L_2-distance, even if they induce the *opposite* ordering of policies.[1] Another option is to evaluate the learnt reward function on a *test set*. However, this is also unsatisfactory, because it can only guarantee that the learnt reward function is accurate on a given data distribution, and when the reward function is *optimised* we necessarily incur a *distributional shift* (after which the learnt reward function may no longer match the true reward function). Yet another option is to optimise the learnt reward function, and evaluate the obtained policy according to the true reward function. However, this is also unsatisfactory, both because it is very expensive, and because it makes it difficult to separate issues with the policy optimisation process from issues with the reward learning algorithm. Moreover, because this method is purely empirical, it cannot be used for theoretical work. These issues make it challenging to evaluate reward learning algorithms in a way that is principled and robust. This in turn makes it difficult to anticipate in what situations a reward learning algorithm might fail, or what their failure modes might look like. It also makes it difficult to compare different reward learning algorithms against each other, without getting results that may be heavily dependent on the experimental setup. These issues limit the applicability of reward learning in practice.

In this paper, we introduce STAndardised Reward Comparison (STARC) metrics, which is a family of *pseudometrics* that quantify the difference between reward functions in a principled way. Moreover, we demonstrate that STARC metrics enjoy strong theoretical guarantees. In particular, we show that STARC metrics induce an upper bound on the worst-case regret that can be induced under arbitrary policy optimisation, which means that a small STARC distance guarantees that two reward functions behave in a similar way. Moreover, we also demonstrate that STARC metrics induce a *lower* bound on worst-case regret. This has the important consequence that any reward function distance metric which induces both an upper and a lower bound on worst-case regret must be bilipschitz equivalent to STARC metrics, which in turn means that they (in a certain sense) are unique. In particular, we should not expect to be able to improve on them in any substantial way. In addition to this, we also evaluate STARC metrics experimentally, and demonstrate that their theoretical guarantees translate into compelling empirical performance. STARC metrics are cheap to compute, which means that they can be used for empirical evaluation of reward learning algorithms. Moreover, they can be calculated from a closed-form expression, which means that they are also suitable for use in theoretical analysis. As such, STARC metrics enable us to evaluate reward learning methods in a way that is both easier and more theoretically principled than relevant alternatives. Our work thus contributes towards building a more rigorous foundation for the field of reward learning.

1.1 RELATED WORK

There are two existing papers that study the problem of how to quantify the difference between reward functions. The first is Gleave et al. (2020), which proposes a distance metric that they call Equivalent-Policy Invariant Comparison (EPIC). They show that the EPIC-distance between two reward functions induces a regret bound for optimal policies. The second paper is Wulfe et al. (2022), which proposes a distance metric that they call Dynamics-Aware Reward Distance (DARD). Unlike EPIC, DARD incorporates information about the transition dynamics of the environment. This means that DARD might give a tighter measurement, in situations where the transition dynamics are known. Unlike Gleave et al. (2020), they do not derive any regret bound for DARD.

Our work extends the work by Gleave et al. (2020) and Wulfe et al. (2022) in several important ways. First of all, Wulfe et al. (2022) do not provide any regret bounds, which is unsatisfactory for theoretical work, and the upper regret bound that is provided by Gleave et al. (2020) is both weaker and less general than ours. In particular, their bound only considers optimal policies, whereas our bound covers all pairs of policies (with optimal policies being a special case). Moreover, we also argue that Gleave et al. (2020) have chosen to quantify regret in a way that fails to capture what we care about in practice. In Appendix A, we provide an extensive theoretical analysis of EPIC, and show

---

[1] For example, given an arbitrary reward function R and an arbitrary constant c, we have that R and $c \cdot R$ have the same ordering of policies, even though their L_2-distance may be arbitrarily large. Similarly, for any ϵ, we have that $\epsilon \cdot R$ and $-\epsilon \cdot R$ have the opposite ordering of policies, unless R is constant, even though their L_2-distance may be arbitrarily small.

that it lacks many of the important theoretical guarantees enjoyed by STARC metrics. In particular, we demonstrate that EPIC fails to induce either an upper or lower bound on worst-case regret (as we define it). We also include an extensive discussion and criticism of DARD in Appendix B. Moreover, in Section 4, we provide experimental data that shows that STARC metrics in practice can have a much tighter correlation with worst-case regret than both EPIC and DARD. This means that STARC metrics both can attain better empirical performance *and* give stronger theoretical guarantees than the pseudometrics proposed by earlier work.

It is important to note that EPIC is designed to be independent of the environment dynamics, whereas both STARC and DARD depend on the transition dynamics. This issue is discussed in Section 2.3.

The question of what happens if one reward function is optimised instead of a different reward function is considered by many previous works. A notable example is Ng et al. (1999), which shows that if two reward functions differ by a type of transformation they call *potential shaping*, then they have the same optimal policies in all environments. Potential shaping is also studied by e.g. Jenner et al. (2022). Another example is Skalse et al. (2022b), which shows that if two reward functions R_1, R_2 have the property that there are no policies π_1, π_2 such that $J_1(\pi_1) > J_1(\pi_2)$ and $J_2(\pi_1) < J_2(\pi_2)$, then either R_1 and R_2 induce the same ordering of policies, or at least one of them assigns the same reward to all policies. Zhuang & Hadfield-Menell (2021) consider proxy rewards that depend on a strict subset of the features which are relevant to the true reward, and then show that optimising such a proxy in some cases may be arbitrarily bad, given certain assumptions. Skalse et al. (2022a) derive necessary and sufficient conditions for when two reward functions are equivalent, for the purposes of computing certain policies or other mathematical objects. Also relevant is Everitt et al. (2017), which studies the related problem of reward corruption, and Pan et al. (2022), which considers natural choices of proxy rewards for several environments. Unlike these works, we are interested in the question of *quantifying* the difference between reward functions.

1.2 PRELIMINARIES

A *Markov Decision Processes* (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, \tau, \mu_0, R, \gamma)$ where $\mathcal{S}$ is a set of *states*, $\mathcal{A}$ is a set of *actions*, $\tau : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a *transition function*, $\mu_0 \in \Delta(\mathcal{S})$ is an *initial state distribution*, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is a *reward function*, and $\gamma \in (0, 1)$ is a *discount rate*. A *policy* is a function $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. A *trajectory* $\xi = \langle s_0, a_0, s_1, a_1 \dots \rangle$ is a possible path in an MDP. The *return function* G gives the cumulative discounted reward of a trajectory, $G(\xi) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})$, and the *evaluation function* J gives the expected trajectory return given a policy, $J(\pi) = \mathbb{E}_{\xi \sim \pi}[G(\xi)]$. A policy maximising J is an *optimal policy*. The *value function* $V^\pi : \mathcal{S} \to \mathbb{R}$ of a policy encodes the expected future discounted reward from each state when following that policy. We use $\mathcal{R}$ to refer to the set of all reward functions. When talking about multiple rewards, we give each reward a subscript R_i, and use J_i, G_i, and V_i^π, to denote R_i's evaluation function, return function, and π-value function.

In this paper, we assume that all states are reachable under τ and μ_0. Note that if this is not the case, then all unreachable states can simply be removed from $\mathcal{S}$. Our *theoretical results* also assume that $\mathcal{S}$ and $\mathcal{A}$ are finite. However, STARC metrics can still be computed in continuous environments.

Given a set X, a function $d : X \times X \to \mathbb{R}$ is called a *pseudometric* if $d(x_1, x_1) = 0$, $d(x_1, x_2) \geqslant 0$, $d(x_1, x_2) = d(x_2, x_1)$, and $d(x_1, x_3) \leqslant d(x_1, x_2) + d(x_2, x_3)$, for all $x_1, x_2, x_2 \in X$. Given two pseudometrics d_1, d_2 on X, if there are constants ℓ, u such that $\ell \cdot d_1(x_1, x_2) \leqslant d_2(x_1, x_2) \leqslant u \cdot d_1(x_1, x_2)$ for all $x_1, x_2 \in X$, then d_1 and d_2 are *bilipschitz equivalent*. Given a vector space V, a function $n : V \to \mathbb{R}$ is a *norm* if $n(v_1) \geqslant 0$, $n(v_1) = 0 \iff v_1 = 0$, $n(c \cdot v_1) = |c| \cdot n(v_1)$, and $n(v_1 - v_2) \leqslant n(v_1) + n(v_2)$ for all $v_1, v_2 \in V$, $c \in \mathbb{R}$. Given a norm n, we can define a (pseudo)metric m as $m(x, y) = n(|x - y|)$. In a mild abuse of notation, we will often denote this metric using n directly, so that $n(x, y) = n(|x - y|)$. For any $p \in \mathbb{N}$, L_p is the norm given by $L_p(v) = (\sum |v_i|^p)^{1/p}$. A norm n is a *weighted* version of n' if $n = n' \circ M$ for a diagonal matrix M.

We will use *potential shaping*, which was first introduced by Ng et al. (1999). First, a *potential function* is a function $\Phi : \mathcal{S} \to \mathbb{R}$. Given a discount γ, we say that R_1 and R_2 differ by *potential shaping* if for some potential Φ, we have that $R_2(s, a, s') = R_1(s, a, s') + \gamma \cdot \Phi(s') - \Phi(s)$. We also use *$S'$-redistribution* (as defined by Skalse et al., 2022a). Given a transition function τ, we say that R_1 and R_2 differ by S'-redistribution if $\mathbb{E}_{S' \sim \tau(s,a)}[R_2(s, a, S')] = \mathbb{E}_{S' \sim \tau(s,a)}[R_1(s, a, S')]$. Finally, we say that R_1 and R_2 differ by positive linear scaling if $R_2(s, a, s') = c \cdot R_1(s, a, s')$ for some positive constant c. We will also combine these transformations. For example, we say that R_1

and R_2 differ by potential shaping and S'-redistribution if it is possible to produce R_2 from R_1 by applying potential shaping and S'-redistribution (in any order). The cases where R_1 and R_2 differ by (for example) potential shaping and positive linear scaling, etc, are defined analogously. Finally, we will use the following result, proven by Skalse & Abate (2023) in their Theorem 2.6:

Proposition 1. *$(S, A, \tau, \mu_0, R_1, \gamma)$ and $(S, A, \tau, \mu_0, R_2, \gamma)$ have the same ordering of policies if and only if R_1 and R_2 differ by potential shaping, positive linear scaling, and S'-redistribution.*

The "ordering of policies" is the ordering induced by the policy evaluation function J.

EPIC (Gleave et al., 2020) is defined relative to a distribution $\mathcal{D}_\mathcal{S}$ over $\mathcal{S}$ and a distribution $\mathcal{D}_\mathcal{A}$ over $\mathcal{A}$, which must give support to all states and actions. It is computed in several steps. First, let $C^{\text{EPIC}} : \mathcal{R} \to \mathcal{R}$ be the function where $C^{\text{EPIC}}(R)(s, a, s')$ is equal to

$$R(s, a, s') + \mathbb{E}[\gamma R(s', A, S') - R(s, A, S') - \gamma R(S, A, S')],$$

where $S, S' \sim \mathcal{D}_\mathcal{S}$ and $A \sim \mathcal{D}_\mathcal{A}$. Note that S and S' are sampled independently. Next, let the "Pearson distance" between two random variables X and Y be defined as $\sqrt{(1 - \rho(X, Y))/2}$, where ρ denotes the Pearson correlation. Then the EPIC-distance $D^{\text{EPIC}}(R_1, R_2)$ is defined to be the Pearson distance between $C^{\text{EPIC}}(R_1)(S, A, S')$ and $C^{\text{EPIC}}(R_2)(S, A, S')$, where again $S, S' \sim \mathcal{D}_\mathcal{S}$ and $A \sim \mathcal{D}_\mathcal{A}$.[2] Note that D^{EPIC} is implicitly parameterised by $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{A}$.

To better understand how EPIC works, it is useful to know that it can be equivalently expressed as

$$D^{\text{EPIC}}(R_1, R_2) = \frac{1}{2} \cdot L_{2,\mathcal{D}} \left(\frac{C^{\text{EPIC}}(R_1)}{L_{2,\mathcal{D}}(C^{\text{EPIC}}(R_1))}, \frac{C^{\text{EPIC}}(R_2)}{L_{2,\mathcal{D}}(C^{\text{EPIC}}(R_2))} \right),$$

where $L_{2,\mathcal{D}}$ is a weighted L_2-norm. For details, see Appendix E. Here C^{EPIC} maps all reward functions that differ by potential shaping to a single representative in their equivalence class. This, combined with the normalisation step, ensures that reward functions which only differ by potential shaping and positive linear scaling have distance 0 under D^{EPIC}.

DARD (Wulfe et al., 2022) is also defined relative to a distribution $\mathcal{D}_\mathcal{S}$ over $\mathcal{S}$ and a distribution $\mathcal{D}_\mathcal{A}$ over $\mathcal{A}$, which must give support to all actions and all reachable states, but it also requires a transition function τ. Let $C^{\text{DARD}} : \mathcal{R} \to \mathcal{R}$ be the function where $C^{\text{DARD}}(R)(s, a, s')$ is

$$R(s, a, s') + \mathbb{E}[\gamma R(s', A, S'') - R(s, A, S') - \gamma R(S', A, S'')],$$

where $A \sim \mathcal{D}_\mathcal{A}$, $S' \sim \tau(s, A)$, and $S'' \sim \tau(s', A)$. Then the DARD-distance $D^{\text{DARD}}(R_1, R_2)$ is defined to be the Pearson distance between $C^{\text{DARD}}(R_1)(S, A, S')$ and $C^{\text{DARD}}(R_2)(S, A, S')$, where again $S, S' \sim \mathcal{D}_\mathcal{S}$ and $A \sim \mathcal{D}_\mathcal{A}$. Note that D^{DARD} is parameterised by $\mathcal{D}_\mathcal{S}$, $\mathcal{D}_\mathcal{A}$, and τ.

2 STARC METRICS

In this section we formally define STARC metrics, and provide several examples of such metrics.

2.1 A FORMAL DEFINITION OF STARC METRICS

STARC metrics are defined relative to an environment, consisting of a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a transition function τ, an initial state distribution μ_0, and a discount factor γ. This means that many of our definitions and theorems are implicitly parameterised by these objects, even when this dependency is not spelled out explicitly. Our results hold for any choice of $\mathcal{S}$, $\mathcal{A}$, τ, μ_0, and γ, as long as they satisfy the assumptions given in Section 1.2. See also Section 2.3.

STARC metrics are computed in several steps, where the first steps collapse certain equivalence classes in $\mathcal{R}$ to a single representative, and the last step measures a distance. The reason for this is that two distinct reward functions can share the exact same preferences between all policies. When this is the case, we want them to be treated as equivalent. This is achieved by standardising the reward functions in various ways before the distance is finally measured. First, recall that neither potential shaping nor S'-redistribution affects the policy ordering in any way. This motivates the first step:

---

[2] Gleave et al. (2020) allow different distributions to be used when computing $C^{\text{EPIC}}(R)$ and when taking the Pearson distance. However, doing this breaks some of their theoretical results. For details, see Appendix E.

Definition 1. A function $c : \mathcal{R} \to \mathcal{R}$ is a *canonicalisation function* if c is linear, $c(R)$ and R only differ by potential shaping and S'-redistribution for all $R \in \mathcal{R}$, and for all $R_1, R_2 \in \mathcal{R}$, $c(R_1) = c(R_2)$ if and only if R_1 and R_2 only differ by potential shaping and S'-redistribution.

Note that we require c to be linear. Note also that C^{EPIC} and C^{DARD} are not canonicalisation functions in our sense, because we here require canonicalisation functions to simultaniously standardise both potential shaping and S'-redistribution, whereas C^{EPIC} and C^{DARD} only standardise potential shaping. In Section 2.2, we provide examples of canonicalisation functions. Let us next introduce the functions that we use to compute a distance:

Definition 2. A metric $m : \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ is *admissible* if there exists a norm p and two (positive) constants u, ℓ such that $\ell \cdot p(x, y) \leqslant m(x, y) \leqslant u \cdot p(x, y)$ for all $x, y \in \mathcal{R}$.

A metric is admissible if it is bilipschitz equivalent to a norm. Any norm is an admissible metric, though there are admissible metrics which are not norms.[3] Recall also that all norms are bilipschitz equivalent on any finite-dimensional vector space. This means that if m satisfies Definition 2 for one norm, then it satisfies it for all norms. We can now define our class of reward metrics:

Definition 3. A function $d : \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ is a *STARC metric* (STAndardised Reward Comparison) if there is a canonicalisation function c, a function n that is a norm on $\mathrm{Im}(c)$, and a metric m that is admissible on $\mathrm{Im}(s)$, such that $d(R_1, R_2) = m(s(R_1), s(R_2))$, where $s(R) = c(R)/n(c(R))$ when $n(c(R)) \neq 0$, and $c(R)$ otherwise.

Intuitively speaking, c ensures that all reward functions which differ by potential shaping and S'-redistribution are considered to be equivalent, and division by n ensures that positive scaling is ignored as well. Note that if $n(c(R)) = 0$, then $c(R)$ assigns 0 reward to every transition. Note also that $\mathrm{Im}(c)$ is the image of c, if c is applied to the entirety of $\mathcal{R}$. If n is a norm on $\mathcal{R}$, then n is also a norm on $\mathrm{Im}(c)$, but there are functions which are norms on $\mathrm{Im}(c)$ but not on $\mathcal{R}$ (c.f. Proposition 4).

In Appendix C, we provide a geometric intuition for how STARC metrics work.

2.2 EXAMPLES OF STARC METRICS

In this section, we give several examples of STARC metrics. We begin by showing how to construct canonicalisation functions. We first give a simple and straightforward method:

Proposition 2. *For any policy π, the function $c : \mathcal{R} \to \mathcal{R}$ given by*

$$c(R)(s, a, s') = \mathbb{E}_{S' \sim \tau(s,a)}\left[R(s, a, S') - V^\pi(s) + \gamma V^\pi(S')\right]$$

is a canonicalisation function. Here V^π is computed under the reward function R given as input to c. We call this function Value-Adjusted Levelling (VAL).

The proof, as well as all other proofs, are given in the Appendix. Proposition 2 gives us an easy way to make canonicalisation functions, which are also easy to evaluate whenever V^π is easy to approximate. We next give another example of canonicalisation functions:

Definition 4. A canonicalisation function c is *minimal* for a norm n if for all R we have that $n(c(R)) \leqslant n(R')$ for all R' such that R and R' only differ by potential shaping and S'-redistribution.

Minimal canonicalisation functions give rise to tighter regret bounds (c.f. Section 3 and Appendix F). It is not a given that minimal canonicalisation functions exist for a given norm n, or that they are unique. However, for any weighted L_2-norm, this is the case:

Proposition 3. *For any weighted L_2-norm, a minimal canonicalisation function exists and is unique.*

A STARC metric can use any canonicalisation function c. Moreover, the normalisation step can use any function n that is a norm on $\mathrm{Im}(c)$. This does of course include the L_1-norm, L_2-norm, L_∞-norm, and so on. We next show that $\max_\pi J(\pi) - \min_\pi J(\pi)$ also is a norm on $\mathrm{Im}(c)$:

Proposition 4. *If c is a canonicalisation function, then the function $n : \mathcal{R} \to \mathcal{R}$ given by $n(R) = \max_\pi J(\pi) - \min_\pi J(\pi)$ is a norm on $\mathrm{Im}(c)$.*

---

[3] For example, the unit ball of m does not have to be convex, or symmetric around the origin.

For the final step we of course have that any norm is an admissible metric, though some other metrics are admissible as well.[4] To obtain a STARC metric, we then pick any canonicalisation function c, norm n, and admissible metric m, and combine them as described in Definition 3. Which choice of c, n, and m is best in a given situation may depend on multiple considerations, such as how easy they are to compute, how easy they are to work with theoretically, or how well they together track worst-case regret (c.f. Section 3 and 4).

2.3 UNKNOWN TRANSITION DYNAMICS AND CONTINUOUS ENVIRONMENTS

STARC metrics depend on the transition function τ, through the definition of canonicalisation functions (since S'-redistribution depends on τ). Moreover, τ is often unknown in practice. However, it is important to note that while STARC metrics *depend* on τ, there are STARC metrics that can be computed without *direct access* to τ. For example, the VAL canonicalisation function (Proposition 2) only requires that we can *sample* from τ, which is always possible in the reinforcement learning setting. Moreover, if we want to evaluate a learnt reward function in an environment that is different from the training environment, then we can simply use the τ from the evaluation environment. As such, we do not consider the dependence on τ to be a meaningful limitation. Nonetheless, it is possible to define STARC-like pseudometrics that do not depend on τ at all, and such pseudometrics also have some theoretical guarantees (albeit guarantees that are weaker than those enjoyed by STARC metrics). This option is discussed in Appendix F.3.

Moreover, we assume that $\mathcal{S}$ and $\mathcal{A}$ are finite, but many interesting environments are *continuous*. However, it is important to note that while our theoretical results assume that $\mathcal{S}$ and $\mathcal{A}$ are finite, it is still straightforward to compute and use STARC metrics in continuous environments (for example, using the VAL canonicalisation function from Proposition 2). We discuss this issue in more detail in Appendix D. In Section 4, we also provide experimental data from a continuous environment.

3 THEORETICAL RESULTS

In this section, we prove that STARC metrics enjoy several desirable theoretical guarantees. First, we note that all STARC metrics are pseudometrics on the space of all reward functions, $\mathcal{R}$:

Proposition 5. *All STARC metrics are pseudometrics on* $\mathcal{R}$*.*

This means that STARC metrics give us a well-defined notion of a "distance" between rewards. Next, we characterise the cases when STARC metrics assign two rewards a distance of zero:

Proposition 6. *All STARC metrics have the property that* $d(R_1, R_2) = 0$ *if and only if* R_1 *and* R_2 *induce the same ordering of policies.*

This means that STARC metrics consider two reward functions to be equivalent, exactly when those reward functions induce exactly the same ordering of policies. This is intuitive and desirable.

For a pseudometric d on $\mathcal{R}$ to be useful, it is crucial that it induces an upper bound on worst-case regret. Specifically, we want it to be the case that if $d(R_1, R_2)$ is small, then the impact of using R_2 instead of R_1 should also be small. When a pseudometric has this property, we say that it is *sound*:

Definition 5. A pseudometric d on $\mathcal{R}$ is *sound* if there exists a positive constant U, such that for any reward functions R_1 and R_2, if two policies π_1 and π_2 satisfy that $J_2(\pi_2) \geqslant J_2(\pi_1)$, then

$$J_1(\pi_1) - J_1(\pi_2) \leqslant U \cdot (\max_\pi J_1(\pi) - \min_\pi J_1(\pi)) \cdot d(R_1, R_2).$$

Let us unpack this definition. $J_1(\pi_1) - J_1(\pi_2)$ is the regret, as measured by R_1, of using policy π_2 instead of π_1. Division by $\max_\pi J_1(\pi) - \min_\pi J_1(\pi)$ normalises this quantity based on the total range of R_1 (though the term is put on the right-hand side of the inequality, instead of being used as a denominator, in order to avoid division by zero when $\max_\pi J_1(\pi) - \min_\pi J_1(\pi) = 0$). The condition that $J_2(\pi_2) \geqslant J_2(\pi_1)$ says that R_2 prefers π_2 over π_1. Taken together, this means that a pseudometric d on $\mathcal{R}$ is sound if $d(R_1, R_2)$ gives an upper bound on the maximal regret that could

---

[4] For example, if $m(x, y)$ is the *angle* between x and y when $x, y \neq 0$, and we define $m(0, 0) = 0$ and $m(x, 0) = \pi/2$ for $x \neq 0$, then m is also admissible, even though m is not a norm.

be incurred under R_1 if an arbitrary policy π_1 is optimised to another policy π_2 according to R_2. It is also worth noting that this includes the special case when π_1 is optimal under R_1 and π_2 is optimal under R_2. Our first main result is that all STARC metrics are sound:

Theorem 1. *All STARC metrics are sound.*

This means that any STARC metric gives us an upper bound on worst-case regret. Next, we will show that STARC metrics also induce a *lower* bound on worst-case regret. It may not be immediately obvious why this property is desirable. To see why this is the case, note that if a pseudometric d on $\mathcal{R}$ does not induce a lower bound on worst-case regret, then there are reward functions that have a *low* worst-case regret, but a *large* distance under d. This would in turn mean that d is not *tight*, and that it should be possible to improve upon it. In other words, if we want a small distance under d to be both sufficient *and necessary* for low worst-case regret, then d must induce both an upper *and a lower* bound on worst-case regret. As such, we also introduce the following definition:

Definition 6. A pseudometric d on $\mathcal{R}$ is *complete* if there exists a positive constant L, such that for any reward functions R_1 and R_2, there exist two policies π_1 and π_2 such that $J_2(\pi_2) \geqslant J_2(\pi_1)$ and

$$J_1(\pi_1) - J_1(\pi_2) \geqslant L \cdot (\max_{\pi} J_1(\pi) - \min_{\pi} J_1(\pi)) \cdot d(R_1, R_2),$$

and moreover, if both $\max_{\pi} J_1(\pi) - \min_{\pi} J_1(\pi) = 0$ and $\max_{\pi} J_2(\pi) - \min_{\pi} J_2(\pi) = 0$, then we have that $d(R_1, R_2) = 0$.

The last condition is included to rule out certain pathological edge-cases. Intuitively, if d is sound, then a small d is *sufficient* for low regret, and if d is complete, then a small d is *necessary* for low regret. Soundness implies the absence of false positives, and completeness the absence of false negatives. Our second main result is that all STARC metrics are complete:

Theorem 2. *All STARC metrics are complete.*

Theorems 1 and 2 together imply that, for any STARC metric d, we have that a small value of d is both necessary and sufficient for a low regret. This means that STARC metrics, in a certain sense, exactly capture what it means for two reward functions to be similar, and that we should not expect it to be possible to significantly improve upon them. We can make this claim formal as follows:

Proposition 7. *Any pseudometrics on $\mathcal{R}$ that are both sound and complete are bilipschitz equivalent.*

This implies that all STARC metrics are bilipschitz equivalent. Moreover, any other pseudometric on $\mathcal{R}$ that induces both an upper and a lower bound on worst-case regret (as we define it) must also be bilipschitz equivalent to STARC metrics.

In Appendix A and B, we provide an extensive analysis of both EPIC and DARD, and show that they fail to induce similar theoretical guarantees.

4 EXPERIMENTAL RESULTS

In this section we present our experimental results. First, we demonstrate that STARC metrics provide a better estimate of regret than EPIC and DARD in randomly generated MDPs. We then evaluate a STARC metric in a continuous environment.

4.1 LARGE NUMBERS OF SMALL RANDOM MDPS

Our first experiment compares several STARC metrics to EPIC, DARD, and a number of other non-STARC baselines. In total, our experiment covered 223 different pseudometrics (including rollout regret), derived by creating different combinations of canonicalisation functions, normalisations, and distance metrics. For details, see Appendix G.3. For each pseudometric, we generated a large number of random MDPs, and then measured how well the pseudometric correlates with *regret* across this distribution. The regret is defined analogously to Definition 5 and 6, except that only optimal policies are considered – for details, see Appendix G.2. We used MDPs with 32 states, 4 actions, $\gamma = 0.95$, a uniform initial state distribution, and randomly sampled sparse non-deterministic transition functions, and for each MDP, we generated several random reward functions. For details on the random generation process, see Appendix G. We compared 49,152 reward function pairs (Appendix G.4), and used these to estimate how well each pseudometric correlates with regret. We show these correlations in Figure 1, and the full data is given in a table in Appendix H. In Appendix H.1, we also provide tables that indicate the impact of changing the metric m or the normalisation function n.

The canonicalisation functions we used were `None` (which simply skips the canonicalisation step), C^{EPIC}, C^{DARD}, `MinimalPotential` (which is the minimal "canonicalisation" that removes potential shaping but not S'-redistribution, and therefore is easier to compute), `VALPotential` (which is given by $R(s,a,s') - V^{\pi}(s) + \gamma V^{\pi}(s')$), and `VAL` (defined in Proposition 2). For both C^{EPIC} and C^{DARD}, both $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{A}}$ were chosen to be uniform over $\mathcal{S}$ and $\mathcal{A}$. For both `VALPotential` and `VAL`, π was chosen to be the uniformly random policy. Note that `VAL` is the only canonicalisation which removes both potential shaping and S'-redistribution, and thus the only one that meets Definition 1 — for this reason, it is listed as "STARC-VAL" in Figure 1. For the full details about which pseudometrics were chosen, and why, see Appendix G.3

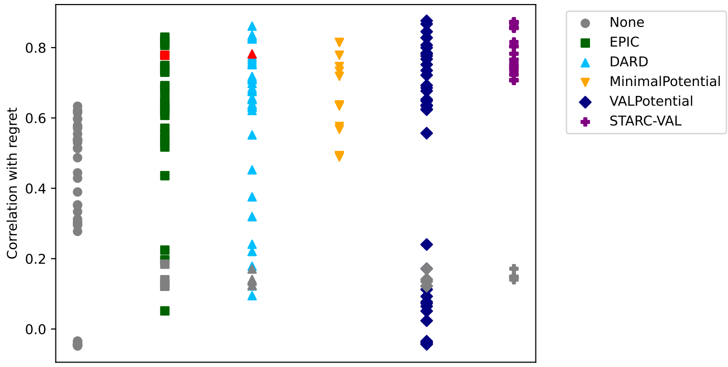

Figure 1: This figure displays the correlation to regret for several pseudometrics. Each point represents one pseudometric, i.e. one unique combination of canonicalisation c, normalisation n, and distance metric m. They are grouped together based on their canonicalisation function, with each column corresponding to a different canonicalisation function. Pseudometrics which skip canonicalisation or normalisation are shown in grey. The versions of EPIC and DARD that use the L_2 norm for both normalisation n and distance metric m are highlighted in red, as these are the original versions given in Gleave et al. (2020) and Wulfe et al. (2022). The STARC metrics, which are canonicalised using VAL, are reliably better indicators of regret than the other pseudometrics.

As we can see, the STARC metrics based on `VAL` perform noticeably better than all pre-existing pseudometrics – for instance, the correlation of EPIC to regret is 0.778, DARD's correlation is 0.782, while `VAL`'s correlation is 0.856 (when using L_2 for both n and m, which is the same as EPIC and DARD). Out of the 10 best pseudometrics, 8 use `VAL` (and the other 2 both use `VALPotential`). Moreover, for each choice of n and m, we have that the `VAL` canonicalisation performs better than the EPIC canonicalisation in 40 out of 42 cases.[5] Taken together, these results suggest that STARC metrics robustly perform better than the existing alternatives.

Our results also suggest that the choice of normalisation function n and metric m can have a significant impact on the pseudometric's accuracy. For instance, when canonicalising with `VAL`, it is better to use the L_1 norm than the L_2 norm for both normalisation and taking the distance – this increases the correlation with regret from 0.856 to 0.873. Another example is the EPIC canonicalisation – when paired with the weighted L_∞ norm for normalisation and the (unweighted) L_∞ norm for taking the distance, instead of using the L_2 norm for both, its correlation decreases from 0.778 to 0.052. As we can see in Figure 1, this effect appears to be more prominent for the non-STARC metrics. Another thing to note is that it seems like `VALPotential` can perform as well as `VAL` despite not canonicalising for S'-redistribution, but only when a (τ-)weighted norm is used. This may be because τ-weighted norms set all impossible transitions to 0, and reduce the impact of very unlikely transitions; plausibly, this could in practice be similar to canonicalising for S'-redistribution. When using `VAL`, L_1 was the best unweighted norm for both m and n in our experiment.

---

[5] The only exceptions are when no normalisation is used and $m = L_\infty$, and when $n =$ `weighted-`L_2 and $m =$ `weighted-`L_∞. However, in the first case, both the EPIC-based and the VAL-based pseudometric perform badly (since no normalisation is used), and in the second case, the difference between them is not large.

4.2 THE REACHER ENVIRONMENT

Our next experiment estimates the distance between several hand-crafted reward functions in the Reacher environment from MuJoCo (Todorov et al., 2012). This is a deterministic environment with an 11-dimensional continuous state space and a 2-dimensional continuous action space. The reward functions we used are:

1. `GroundTruth`: The Euclidean distance to the target, plus a penalty term for large actions.
2. `PotentialShaped`: `GroundTruth` with random potential shaping.
3. `SecondPeak`: We create a second target in the environment, and reward the agent based on both its distance to this target, and to the original target, but give a greater weight to the original target.
4. `Random`: A randomly generated reward, implemented as an affine transformation from s, a, s' to real numbers with the weights and bias randomly initialised.
5. `Negative`: Returns $-$`GroundTruth`.

We expect `GroundTruth` to be equivalent to `PotentialShaped`, similar to `SecondPeak`, orthogonal to `Random`, and opposite to `Negative`. We used the VAL canonicalisation function with the uniform policy, and normalised and took the distance with the L_2-norm. This pseudometric was then estimated through sampling; full details can be found in Appendix D and I. The results of this experiment are given in Table 1. As we can see, the relative ordering of the reward functions match what we expect. However, the magnitudes of the estimated distances are noticeably larger than their real values; for example, the actual distance between `GroundTruth` and `PotentialShaped` is 0, but it is estimated as ≈ 0.9. The reason for this is likely that the estimation involves summing over absolute values, which makes all noise positive. Nonetheless, for the purposes of *ranking* the rewards, this is not fundamentally problematic.

`PotentialShaped`	`SecondPeak`	`Random`	`Negative`
0.8968	1.2570	1.3778	1.706

Table 1: This figure displays the estimated distance (using $c =$ VAL, $n = L_2$, and $m = L_2$) between each reward function in the Reacher environment and the `GroundTruth` reward function.

5 DISCUSSION

We have introduced STARC metrics, and demonstrated that they provide compelling theoretical guarantees. In particular, we have shown that they are both sound and complete, which means that they induce both an upper and a lower bound on worst-case regret. As such, a small STARC distance is both necessary and sufficient to ensure that two reward functions induce a similar ordering of policies. Moreover, any two pseudometrics that are both sound and complete must be bilipschitz equivalent. This means that any pseudometric on $\mathcal{R}$ that has the same theoretical guarantees as STARC metrics must be equivalent to STARC metrics. This means that we have provided what is essentially a complete answer to the question of how to correctly measure the distance between reward functions. Moreover, our experiments show that STARC metrics have a noticeably better empirical performance than any existing pseudometric in the current literature, for a wide range of environments. This means that STARC metrics offer direct practical advantages, in addition to their theoretical guarantees. In addition to this, STARC metrics are both easy to compute, and easy to work with mathematically. As such, STARC metrics will be useful for both empirical and theoretical work on the analysis and evaluation of reward learning algorithms.

Our work can be extended in a number of ways. First of all, it would be desirable to establish more conclusively which STARC metrics work best in practice. Our experiments are indicative, but not conclusive. Secondly, our theoretical results assume that $\mathcal{S}$ and $\mathcal{A}$ are finite; it would be desirable to generalise them to continuous environments. Third, we use a fairly strong definition of regret. We could consider some weaker criterion, that may allow for the creation of more permissive reward metrics. Finally, our work considers the MDP setting – it would be interesting to also consider other classes of environments. We believe that the multi-agent setting would be of particular interest, since it introduces new and more complex dynamics that are not present in the case of MDPs.

REFERENCES

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2017.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. Reinforcement learning with a corrupted reward channel. *CoRR*, abs/1705.08417, 2017. URL `http://arxiv.org/abs/1705.08417`.

Eugene A. Feinberg and Uriel G. Rothblum. Splitting randomized stationary policies in total-reward markov decision processes. *Mathematics of Operations Research*, 37(1):129–153, 2012. ISSN 0364765X, 15265471. URL `http://www.jstor.org/stable/41412346`.

Adam Gleave, Michael Dennis, Shane Legg, Stuart Russell, and Jan Leike. Quantifying differences in reward functions, 2020. URL `https://arxiv.org/abs/2006.13900`.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in Atari. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, volume 31, pp. 8022–8034, Montréal, Canada, 2018. Curran Associates, Inc., Red Hook, NY, USA.

Erik Jenner, Herke van Hoof, and Adam Gleave. Calculus on MDPs: Potential shaping as a gradient, 2022. URL `https://arxiv.org/abs/2208.09570`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Andrew Y Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, volume 1, pp. 663–670, Stanford, California, USA, 2000. Morgan Kaufmann Publishers Inc.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, Bled, Slovenia, 1999. Morgan Kaufmann Publishers Inc.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models, 2022. URL `https://arxiv.org/abs/2201.03544`.

Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 4 edition, 2020.

Joar Skalse and Alessandro Abate. Misspecification in inverse reinforcement learning, 2023.

Joar Skalse, Matthew Farrugia-Roberts, Stuart Russell, Alessandro Abate, and Adam Gleave. Invariance in policy optimisation and partial identifiability in reward learning. *arXiv preprint arXiv:2203.07475*, 2022a.

Joar Skalse, Niki Howe, Krasheninnikov Dima, and David Krueger. Defining and characterizing reward hacking. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2022b.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018. ISBN 9780262352703.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Blake Wulfe, Logan Michael Ellis, Jean Mercat, Rowan Thomas McAllister, and Adrien Gaidon. Dynamics-aware comparison of learned reward functions. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=CALFyKVs87`.

Simon Zhuang and Dylan Hadfield-Menell. Consequences of misaligned AI. *CoRR*, abs/2102.03896, 2021. URL `https://arxiv.org/abs/2102.03896`.