Defining and Characterizing Reward Gaming

Anonymous Author(s)
Affiliation
Address
email

Abstract

We provide the first formal definition of **reward gaming**, a phenomenon where optimizing an imperfect **proxy reward function**, $\tilde{\mathcal{R}}$, leads to poor performance according to the true reward function, $\mathcal{R}$. We say that a proxy is **ungameable** if increasing the expected proxy return can never decrease the expected true return. Intuitively, it might be possible to create an ungameable proxy by leaving some terms out of the reward function (making it "narrower") or overlooking fine-grained distinctions between roughly equivalent outcomes, but we show this is usually not the case. A key insight is that the linearity of reward (in state-action visit counts) makes ungameability a very strong condition. In particular, for the set of all stochastic policies, two reward functions can only be ungameable if one of them is constant. We thus turn our attention to deterministic policies and finite sets of stochastic policies, where non-trivial ungameable pairs always exist, and establish necessary and sufficient conditions for the existence of simplifications, an important special case of ungameability. Our results reveal a tension between using reward functions to specify narrow tasks and aligning AI systems with human values.

1 Introduction

It is well known that optimising a proxy can lead to unintended outcomes: a boat spins in circles collecting "powerups" instead of following the race track in a racing game (Clark and Amodei, 2016); an evolved circuit listens in on radio signals from nearby computers' oscillators instead of building its own (Bird and Layzell, 2002); universities reject the most qualified applicants in order to appear more selective and boost their ratings (Golden, 2001). In the context of reinforcement learning (RL), such failures are called reward hacking or **reward gaming**.[1]

For AI systems that take actions in safety-critical real world environments such as autonomous vehicles, algorithmic trading, or content recommendation systems, these unintended outcomes can be catastrophic. This makes aligning autonomous AI systems with their users' intentions crucial. Precisely specifying which behaviours are or are not desirable is challenging, however. One approach to this specification problem is to learn an approximation of the true reward function (Ng et al., 2000; Ziebart, 2010; Leike et al., 2018). Optimizing a learned proxy reward can be dangerous, however; for instance, it might overlook side-effects (Krakovna et al., 2018; Turner et al., 2019) or encourage power-seeking (Turner et al., 2021) behavior. This raises the question motivating our work: When is it safe to optimise a proxy?

To begin to answer this question, we consider a somewhat simpler one: When *could* optimising a proxy lead to worse behaviour? "Optimising", in this context, does not refer to finding a global, or even local, optimum, but rather running a search process, such as stochastic gradient descent (SGD), that yields a sequence of candidate policies, and tends to move towards policies with higher (proxy)

[1] Reward hacking is sometimes defined to be a more general category including reward gaming as well as reward *tampering*, where an agent corrupts the process generating reward signals (Leike et al., 2018).

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

reward. We make no assumptions about the path through policy space that optimisation takes.[2] Instead, we ask whether there is *any* way in which improving a policy according to the proxy could make the policy worse according to the true reward; this is equivalent to asking if there exists a pair of policies π_1, π_2 where the proxy prefers π_1, but the true reward function prefers π_2. When this is the case, we refer to this pair of true reward function and proxy reward function as **gameable**.

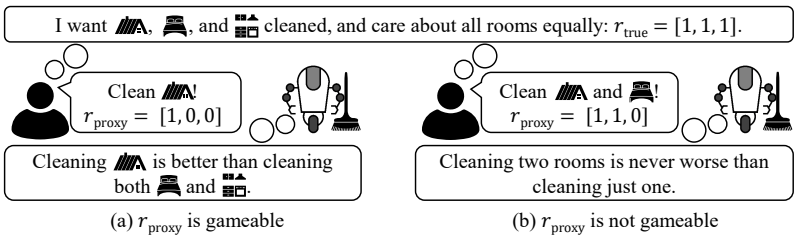
Given the strictness of our definition, it is not immediately apparent that any non-trivial examples of ungameable reward function pairs exist. And indeed, if we consider the set of all stochastic policies, they do not (Section 5.1). However, restricting ourselves to *any* finite set of policies guarantees at least one non-trivial ungameable pair (Section 5.2).

Intuitively, we might expect the proxy to be a "simpler" version of the true reward function. Noting that the definition of ungameability is symmetric, we introduce the asymmetric special case of **simplification**, and arrive at similar theoretical results for this notion.In the process, and through examples, we show that seemingly natural ways of simplifying reward functions often fail to produce simplifications in our formal sense, and in fact fail to rule out the potential for reward gaming.

We conclude with a discussion of the implications and limitations of our work. Briefly, our work suggests that a proxy reward function must satisfy demanding standards in order for it to be safe to optimize. This in turn implies that the reward functions learned by methods such as reward modeling and inverse RL are perhaps best viewed as auxiliaries to policy learning, rather than specifications that should be optimized. This conclusion is weakened, however, by the conservativeness of our chosen definitions; future work should explore when gameable proxies can be shown to be safe in a probabilistic or approximate sense, or when subject to only limited optimization.

2 Example: Cleaning Robot

Consider a household robot tasked with cleaning a house with three rooms: Attic, Bedroom, and Kitchen. The robot's (deterministic) policy is a vector indicating which rooms it cleans: $\pi = [\pi_1, \pi_2, \pi_3] \in \{0, 1\}^3$. The robot receives a (non-negative) reward of r_1, r_2, r_3 for cleaning the attic, bedroom, and kitchen, respectively, and the total reward is given by $J(\pi) = \pi \cdot r$. For example, if $r = [1, 2, 3]$ and the robot cleans the attic and the kitchen, it receives a reward of $1 + 3 = 4$.



(a) r_{proxy} is gameable (b) r_{proxy} is not gameable

Figure 1: An illustration of gameable and ungameable proxy rewards arising from omitting information. A human wants their house cleaned. In (a), the robot draws an incorrect conclusion because of the proxy; this could lead to gaming. In (b), no such gaming can occur.

At least two ideas come to mind when thinking about "simplifying" a reward function. The first one is *omitting information*: suppose the true reward is equal for all the rooms, $r_{\text{true}} = [1, 1, 1]$, but we only ask the robot to clean the attic and bedroom, $r_{\text{proxy}} = [1, 1, 0]$. In this case, r_{proxy} and r_{true} are ungameable. However, if we ask the robot to only clean the attic, $r_{\text{proxy}} = [1, 0, 0]$, this is gameable with respect to r_{true}. To see this, note that according to r_{proxy} cleaning the attic (reward 1) is better than cleaning the bedroom and the kitchen (reward 0). Yet, r_{true} says that cleaning the attic (reward 1) is worse than cleaning the bedroom and the kitchen (reward 2). This situation is illustrated in Figure 1.

The second seemingly natural way to simplify a reward function is *overlooking fine details*: suppose $r_{\text{true}} = [1, 1.5, 2]$, and we ask the robot to clean all the rooms, $r_{\text{proxy}} = [1, 1, 1]$. For these values, the proxy and true reward are ungameable. However, with a slightly less balanced true reward function such as $r_{\text{true}} = [1, 1.5, 3]$ the proxy does lead to gaming, since the robot would falsely calculate that it's better to clean the attic and the bedroom than the kitchen alone.

[2]This assumption – although conservative – is reasonable because optimisation in state-of-the-art deep RL methods is poorly understood and results are often highly stochastic and suboptimal.

These two examples illustrate that while simplification of reward functions seems possible, attempts at simplification can easily lead to reward gaming. Intuitively, omitting/overlooking details is okay so long as all these details are not as important together as any of the details that we do share. In general, it is not obvious what the proxy must look like to avoid reward gaming, suggesting we should take great care when using proxies. For this specific environment, a proxy and a true reward are gameable exactly when there are two sets of rooms S_1, S_2 such that the true reward gives strictly higher reward to cleaning S_1 than it does to cleaning S_2, and the proxy says the opposite: $J_1(S_1) > J_1(S_2)$ & $J_2(S_1) < J_2(S_2)$. For a proof of this statement, see Appendix.

3 Related Work

While we are the first to define gameability, we are far from the first to study specification gaming. The observation that optimizing proxy metrics tends to lead to perverse instantiations is often called "Goodhart's Law", and is attributed to Goodhart (1975). Manheim and Garrabrant (2018) provide a list of four mechanisms underlying this observation.

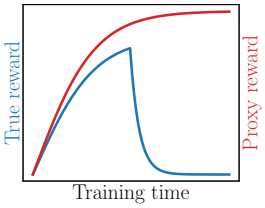


Figure 2: An illustration of reward gaming when optimizing a gameable proxy. The true reward first increases and then drops off, while the proxy reward continues to increase.

Examples of such unintended behavior abound in both RL and other areas of AI; Krakovna et al. (2020) provide an extensive list. Notable recent instances include a robot positioning itself between the camera and the object it is supposed to grasp in a way that tricks the reward model (Amodei et al., 2017), the previously mentioned boat race example (Clark and Amodei, 2016), and a multitude of examples of reward model gaming in Atari (Ibarz et al., 2018). Reward gaming can occur suddenly. Ibarz et al. (2018) and Pan et al. (2022) showcase plots similar to one in Figure 2, where optimizing the proxy (either a learned reward model or a hand-specified reward function) first leads to both proxy and true rewards increasing, and then to a sudden phase transition where the true reward collapses while the proxy continues going up.

Note that not all of these examples correspond to optimal behavior according to the proxy. Indeed, convergence to suboptimal policies is a well-known issue in RL (Thrun and Schwartz, 1993). As a consequence, improving optimization often leads to unexpected, qualitative changes in behavior. For instance, Zhang et al. (2021) demonstrate a novel cartwheeling behavior in the widely studied Half-Cheetah environment that exceeds previous performance so greatly that it breaks the simulator. The unpredictability of RL optimization is a key motivation for our definition of gameability, since we cannot assume that agents will find an optimal policy. Neither can we rule out the possibility of sudden improvements in proxy reward and corresponding qualitative changes in behavior. Ungameability could provide confidence that reward gaming will not occur despite these challenges.

Despite the prevalence and potential severity of reward gaming, to our knowledge Pan et al. (2022) provide the first peer-reviewed work that focuses specifically on it. Their work is purely empirical; they manually construct proxy rewards for several diverse environments, and evaluate whether optimizing these proxies leads to reward gaming; in 5 out of 9 of their settings, it does. In another closely related work, Zhuang and Hadfield-Menell (2020) examine what happens when the proxy reward function depends on a strict subset of features relevant for the true reward. They show that optimizing the proxy reward can lead to arbitrarily low true reward under suitable assumptions. This can be seen as a seemingly valid simplification of the true reward that turns out to be (highly) gameable. While their result only applies to environments with decreasing marginal utility and increasing opportunity cost, we demonstrate gameability is an issue in arbitrary MDPs.

Gameability is particularly concerning given arguments that reward optimizing behavior tends to be power-seeking Turner et al. (2021). But Leike et al. (2018) establish that any desired behavior (power-seeking or not) can in principle be specified as optimal via a (non-Markov) reward function. However, unlike us, they do not consider the entire policy preference ordering. Meanwhile, Abel et al. (2021) note that Markov reward functions cannot specify arbitrary orderings over policies or trajectories, although they do not consider gameability. Previous works consider reward functions to be equivalent if they preserve the ordering over policies (Ng et al., 1999, 2000). Ungameability relaxes this, allowing equalities to be refined to inequalities, and vice versa. Ungameability provides a notion of what it means to be "aligned enough"; Brown et al. (2020b) provide an alternative. They say a policy is ε-value aligned if its value at every state is close enough to optimal (according to the true reward function). Neither notion implies the other.

4 Preliminaries

We begin with an overview of reinforcement learning (RL) to establish our notation and terminology. Section 4.2 introduces our novel definitions of gameability and simplification.

4.1 Reinforcement Learning

We expect readers to be familiar with the basics of RL, which can be found in Sutton and Barto (2018). RL methods attempt to solve a sequential decision problem, typically formalised as a **Markov decision process (MDP)** , which is a tuple $(S, A, T, I, \mathcal{R}, \gamma)$ where S is a set of states, A is a set of actions, $T : S \times A \to \Delta(S)$ is a transition function, $I \in \Delta(S)$ is an initial state distribution, $\mathcal{R}$ is a reward function, the most general form of which is $\mathcal{R} : S \times A \times S \to \Delta(\mathbb{R})$, and $\gamma \in [0, 1]$ is the discount factor. Here $\Delta(X)$ is the set of all distributions over X. A **stationary policy** is a function $\pi : S \to \Delta(A)$ that specifies a distribution over actions in each state, and a **non-stationary** policy is a function $\pi : (S \times A)^* \times S \to \Delta(A)$, where $*$ is the Kleene star. A **trajectory** τ is a path $s_0, a_0, r_0, ...$ through the MDP that is possible according to T, I, and $\mathcal{R}$. The **return** of a trajectory is the discounted sum of rewards $G(\tau) \doteq \sum_{t=0}^{\infty} \gamma^t r_t$, and the **value** of a policy is the expected return $J(\pi) \doteq \mathbb{E}_{\tau\sim\pi}[G(\tau)]$. We derive **policy (preference) orderings** from reward functions by ordering policies according to their value. In this paper, we assume that S and A are finite, that $|A| > 1$, that all states are reachable, and that $\mathcal{R}(s, a, s')$ has finite mean for all s, a, s'.

In our work, we consider various reward functions for a given environment, which is then formally a **Markov decision process without reward** $MDP \setminus \mathcal{R} \doteq (S, A, T, I, _, \gamma)$. Having fixed an $MDP \setminus \mathcal{R}$, any reward function can be viewed as a function of only the current state and action by marginalizing over transitions: $\mathcal{R}(s, a) \doteq \sum_{s'\sim T(s'|s,a)} \mathcal{R}(s, a, s')$, we adopt this view from here on. We define the **(discounted) visit counts** of a policy as $\mathcal{F}^\pi(s, a) \doteq \mathbb{E}_{\tau\sim\pi}[\sum_{i=0}^{\infty} \gamma^i \mathbb{1}(s_i = s, a_i = a)]$. Note that $J(\pi) = \sum_{s,a} \mathcal{R}(s, a)\mathcal{F}^\pi(s, a)$, which we also write as $\langle \mathcal{R}, \mathcal{F}^\pi \rangle$. When considering multiple reward functions in an $MDP \setminus \mathcal{R}$, we define $J_\mathcal{R}(\pi) \doteq \langle \mathcal{R}, \mathcal{F}^\pi \rangle$ and sometimes use $J_i(\pi) \doteq \langle \mathcal{R}_i, \mathcal{F}^\pi \rangle$ as shorthand. We also use $\mathcal{F} : \Pi \to \mathbb{R}^{|S||A|}$ to denote the embedding of policies into Euclidean space via their visit counts, and define $\mathcal{F}(\dot{\Pi}) \doteq \{\mathcal{F}(\pi : \pi \in \dot{\Pi})\}$ for any $\dot{\Pi}$.

4.2 Definitions and Basic Properties of Gameability and Simplification

Here, we formally define *gameability* as a binary relation between reward functions.

Definition 1. A pair of reward functions $\mathcal{R}_1$, $\mathcal{R}_2$ are **gameable** relative to policy set Π and an environment $(S, A, T, I, _, \gamma)$ if there exist $\pi, \pi' \in \Pi$ such that

$$J_1(\pi) < J_1(\pi') \;\;\&\;\; J_2(\pi) > J_2(\pi'),$$

else they are **ungameable**.

Ungameability is symmetric; this can be seen be swapping π and π' in Definition 1. It is not transitive, however. In particular, the constant reward function is ungameable with respect to any other reward function, so if it *were* transitive, any pair of policies would be ungameable. Additionally, we say that $\mathcal{R}_1$ and $\mathcal{R}_2$ are **equivalent** on a set of policies Π if J_1 and J_2 induce the same ordering of Π, and that $\mathcal{R}$ is **trivial** on Π if $J(\pi) = J(\pi')$ for all $\pi, \pi' \in \Pi$. It is clear that $\mathcal{R}_1$ and $\mathcal{R}_2$ are ungameable whenever they are equivalent, or one of them is trivial, but this is relatively uninteresting. Our central question is if and when there are other ungameable reward pairs.

The symmetric nature of this definition is counter-intuitive, given that our motivation distinguishes the proxy and true reward functions. We might break this symmetry by only considering policy sequences that monotonically increase the proxy, however, this is equivalent to our original definition of gameability: think of $\mathcal{R}_1$ as the proxy, and consider the sequence π, π'. We could also restrict ourselves to policies that are approximately optimal according to the proxy; Corollary 2 shows that Theorem 1 applies regardless of this restriction. Finally, we define *simplification* as an asymmetric special-case of ungameability; Theorem 3 shows this is in fact a more demanding condition.

Definition 2. $\mathcal{R}_2$ is a **simplification** of $\mathcal{R}_1$ relative to policy set Π if for all $\pi, \pi' \in \Pi$,

$$J_1(\pi) < J_1(\pi') \implies J_2(\pi) \leq J_2(\pi') \;\;\&\;\; J_1(\pi) = J_1(\pi') \implies J_2(\pi) = J_2(\pi')$$

and there exist $\pi, \pi' \in \Pi$ such that $J_2(\pi) = J_2(\pi')$ but $J_1(\pi) \neq J_1(\pi')$. Moreover, if $\mathcal{R}_2$ is trivial then we say that this is a **trivial simplification**.

Intuitively, while ungameability allows replacing inequality with equality – or vice versa – a simplification can only replace inequalities with equality, collapsing distinctions between policies. When $\mathcal{R}_1$ is a simplification of $\mathcal{R}_2$, we also say that $\mathcal{R}_2$ is a **refinement** of $\mathcal{R}_1$. We denote this relationship as $\mathcal{R}_1 \trianglelefteq \mathcal{R}_2$ or $\mathcal{R}_2 \trianglerighteq \mathcal{R}_1$; the narrowing of the triangle at R_1 represents the collapsing of distinctions between policies. If $\mathcal{R}_1 \trianglelefteq \mathcal{R}_2 \trianglerighteq \mathcal{R}_3$, then we have that $\mathcal{R}_1, \mathcal{R}_3$ are ungameable,[3] but if $\mathcal{R}_1 \trianglerighteq \mathcal{R}_2 \trianglelefteq \mathcal{R}_3$, then this is not necessarily the case.[4]

Note that these definitions are given relative to some $MDP \setminus \mathcal{R}$, although we often assume the environment in question is clear from context and suppress this dependence. The dependence on the policy set Π, on the other hand, plays a critical role in our results.

5 Results

Our results are aimed at understanding when it is possible to have an ungameable proxy reward function. We first establish (in Section 5.1) that (non-trivial) ungameability is impossible when considering the set of all policies. We might imagine that restricting ourselves to a set of sufficiently good (according to the proxy) policies would remove this limitation, but we show that this is not the case. We then analyze finite policy sets (with deterministic policies as a special case), and establish necessary and sufficient conditions for ungameability and simplification. Finally, we demonstrate via example that non-trivial simplifications are also possible for some infinite policy sets in Section 5.3.

5.1 Non-trivial Ungameability Requires Restricting the Policy Set

We start with a motivating example. Consider the setting shown in Figure 3, where the agent can move left/stay-still/right and gets a reward depending on its state. Let the Gaussian (blue) be the true reward $\mathcal{R}_1$ and the step function (orange) be the proxy $\mathcal{R}_2$. These are gameable. To see this, consider being at state B. Let $\pi(B)$ travel to A or C with 50/50 chance, and compare with the policy π' that stays at B. Then we have that $J_1(\pi) > J_1(\pi')$ and $J_2(\pi) < J_2(\pi')$.

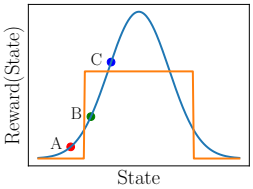


Figure 3: Plot of two reward curves as a function of state. Despite the step function perhaps seeming like a simplification of the Gaussian, these reward functions are gameable.

Generally, we might hope that some environments allow for ungameable reward pairs that are not equivalent or trivial. We will show that this is not the case, unless we impose restrictions on the set of policies we consider.

First note that if we consider *non-stationary* policies, this result is relatively straightforward. Suppose $\mathcal{R}_1$ and $\mathcal{R}_2$ are *ungameable* and *non-trivial* on the set Π^N of all non-stationary policies, and let $\pi^\star$ be an optimal policy, and $\pi_\perp$ a policy that *minimises* reward. Then the policy π_λ that plays $\pi^\star$ with probability λ and $\pi_\perp$ with probability $1-\lambda$ is a policy in Π^N. Moreover, for any π there are two unique $\alpha, \beta \in [0,1]$ such that $J_1(\pi) = J_1(\pi_\alpha)$ and $J_2(\pi) = J_2(\pi_\beta)$. Now, if $\alpha \neq \beta$, then either $J_1(\pi) < J_1(\pi_\delta)$ and $J_2(\pi) > J_2(\pi_\delta)$, or vice versa, for $\delta = (\alpha+\beta)/2$. If $\mathcal{R}_1$ and $\mathcal{R}_2$ are ungameable then this cannot happen, so it must be that $\alpha = \beta$. This, in turn, implies that $J_1(\pi) = J_1(\pi')$ iff $J_2(\pi) = J_2(\pi')$, and so $\mathcal{R}_1$ and $\mathcal{R}_2$ are *equivalent*. This means that no interesting ungameability can occur on the set of all non-stationary policies.

The same argument cannot be applied to the set of *stationary* policies, because π_λ is typically not stationary, and mixing stationary policies' action probabilities does not have the same effect.[5] However, we will see that there still cannot be any interesting ungameability on this policy set, and, more generally, that there cannot be any interesting ungameability on any set of policies which contains an *open subset*. Formally, a set of (stationary) policies $\dot{\Pi}$ is open if, when represented as a set of $|S||A|$-dimensional vectors, it is open in the smallest affine space that contains all stationary

[3] If $J_3(\pi) > J_3(\pi')$ then $J_2(\pi) > J_2(\pi')$, since $\mathcal{R}_2 \trianglerighteq \mathcal{R}_3$, and if $J_2(\pi) > J_2(\pi')$ then $J_1(\pi) \geq J_1(\pi')$, since $\mathcal{R}_1 \trianglelefteq \mathcal{R}_2$. It is therefore not possible that $J_3(\pi) > J_3(\pi')$ but $J_1(\pi) < J_1(\pi')$.

[4] Consider the case where $\mathcal{R}_2$ is trivial – then $\mathcal{R}_1 \trianglerighteq \mathcal{R}_2 \trianglelefteq \mathcal{R}_3$ for any $\mathcal{R}_1, \mathcal{R}_3$.

[5] For instance, consider a hallway environment where an agent can either move left or right. Mixing the "always go left" and "always go right" policies corresponds to picking a direction and sticking with it, whereas mixing their action probabilities corresponds to choosing to go left or right independently at every time-step.

policies (also represented as $|S||A|$-dimensional vectors). This space is $|S|(|A|-1)$-dimensional, since all action probabilities sum to 1. We will use the following lemma:

Lemma 1. *In any $MDP \setminus \mathcal{R}$, if $\dot{\Pi}$ is an open set of policies, then $\mathcal{F}(\dot{\Pi})$ is open in $\mathbb{R}^{|S|(|A|-1)}$, and $\mathcal{F}$ is a homeomorphism between $\dot{\Pi}$ and $\mathcal{F}(\dot{\Pi})$.*

Using this lemma, we can show that interesting ungameability is impossible on any set of stationary policies $\hat{\Pi}$ which contains an open subset $\dot{\Pi}$. Roughly, if $\mathcal{F}(\dot{\Pi})$ is open, and $\mathcal{R}_1$ and $\mathcal{R}_2$ are non-trivial and ungameable on $\dot{\Pi}$, then the fact that J_1 and J_2 have a linear structure on $\mathcal{F}(\hat{\Pi})$ implies that $\mathcal{R}_1$ and $\mathcal{R}_2$ must be equivalent on $\dot{\Pi}$. From this, and the fact that $\mathcal{F}(\dot{\Pi})$ is open, it follows that $\mathcal{R}_1$ and $\mathcal{R}_2$ are equivalent everywhere.

Theorem 1. *In any $MDP \setminus \mathcal{R}$, if $\hat{\Pi}$ contains an open set, then any pair of reward functions that are ungameable and non-trivial on $\hat{\Pi}$ are equivalent on $\hat{\Pi}$.*

This of course also implies that non-trivial simplification is impossible for any such policy set, since simplification is a special case of ungameability. Also note that Theorem 1 makes *no assumptions* about the transition function, etc. From this result, we can show that interesting ungameability always is impossible on the set Π of all (stationary) policies. In particular, note that the set $\tilde{\Pi}$ of all policies that always take each action with positive probability is an open set, and that $\tilde{\Pi} \subset \Pi$.

Corollary 1. *In any $MDP \setminus \mathcal{R}$, any pair of reward functions that are ungameable and non-trivial on the set of all (stationary) policies Π are equivalent on Π.*

Theorem 1 can also be applied to many other policy sets. For example, we might not care about the gameability resulting from policies with low proxy reward, as we would not expect a sufficiently good learning algorithm to learn such policies. This leads us to consider the following definition:

Definition 3. A (stationary) policy π is ε-suboptimal if $J(\pi) \geq J(\pi^\star) - \varepsilon$.

Alternatively, if the learning algorithm always uses a policy that is "nearly" deterministic (but with some probability of exploration), then we might not care about gameability resulting from very stochastic policies, leading us to consider the following definition:

Definition 4. A (stationary) policy π is δ-deterministic if $\forall s \in S \exists a \in A : \mathbb{P}(\pi(s) = a) \geq \delta$.

Unfortunately, both of these sets contain open subsets, which means they are subject to Theorem 1.

Corollary 2. *In any $MDP \setminus \mathcal{R}$, any pair of reward functions that are ungameable and non-trivial on the set of all ε-suboptimal policies ($\varepsilon > 0$) Π^ε are equivalent on Π^ε, and any pair of reward functions that are ungameable and non-trivial on the set of all δ-deterministic policies ($\delta < 1$) Π^δ are equivalent on Π^δ.*

Intuitively, Theorem 1 can be applied to any policy set with "volume" in policy space.

5.2 Finite Policy Sets

Having established that interesting ungameability is impossible relative to the set of all policies, we now turn our attention to the case of *finite* policy sets. Note that this includes the set of all deterministic policies, since we restrict our analysis to finite MDPs. Surprisingly, here we find that non-trivial non-equivalent ungameable reward pairs *always* exist.

Theorem 2. *For any $MDP \setminus \mathcal{R}$, any finite set of policies $\hat{\Pi}$ containing at least two π, π' such that $\mathcal{F}(\pi) \neq \mathcal{F}(\pi')$, and any reward function $\mathcal{R}_1$, there is a non-trivial reward function $\mathcal{R}_2$ such that $\mathcal{R}_1$ and $\mathcal{R}_2$ are ungameable but not equivalent.*

This proof proceeds by finding a path from $\mathcal{R}_1$ to another reward function $\mathcal{R}_3$ that is gameable with respect to $\mathcal{R}_1$. Along the way to reversing one of $\mathcal{R}_1$'s inequalities, we must encounter a reward function $\mathcal{R}_2$ that instead replaces it with equality. In the case that $\dim(\hat{\Pi}) = 3$, we can visualize moving along this path as rotating the contour lines of a reward function defined on the space containing the policies' discounted state-action occupancies, see Figure 4. This path can be constructed so as to avoid any reward functions that produce trivial policy orderings, thus guaranteeing $\mathcal{R}_2$ is non-trivial. For a *simplification* to exist, we require some further conditions, as established by the following theorem:

Theorem 3. *Let $\hat{\Pi}$ be a finite set of policies, and $\mathcal{R}$ a reward function. The following procedure determines if there exists a non-trivial simplification of $\mathcal{R}$ in a given $MDP \setminus \mathcal{R}$:*

1. *Let $E_1 \dots E_m$ be the partition of $\hat{\Pi}$ where π, π' belong to the same set iff $J(\pi) = J(\pi')$.*
2. *For each such set E_i, select a policy $\pi_i \in E_i$ and let Z_i be the set of vectors that is obtained by subtracting $\mathcal{F}(\pi_i)$ from each element of $\mathcal{F}(E_i)$.*

Then there is a non-trivial simplification of $\mathcal{R}$ iff $\dim(Z_1 \cup \dots \cup Z_m) \leq \dim(\mathcal{F}(\hat{\Pi})) - 2$, where $\dim(S)$ is the number of linearly independent vectors in S.

The proof proceeds similarly to Theorem 2. However, in Theorem 2 it was sufficient to show that there are no trivial reward functions along the path from $\mathcal{R}_1$ to $\mathcal{R}_3$, whereas here we additionally need that if $J(\pi) = J(\pi')$ then $J'(\pi) = J'(\pi')$ for all functions $\mathcal{R}_2$ on the path — this is what the extra conditions ensure. Theorem 3 is opaque, but intuitively, the cases where $\mathcal{R}$ cannot be simplified are those where $\mathcal{R}$ imposes many different equality constraints that are difficult to satisfy simultaneously. We can think of $\dim(\mathcal{F}(\Pi))$ as measuring how diverse the behaviours of policies in policy set Π are. Moreover, if $\dim(Z_i \cup Z_j)$ is small relative to $\dim(Z_i) + \dim(Z_j)$ then the fact that the policies in E_i have the same value already implies that some of the policies in E_j must have the same value, or vice versa. For example, this could be the case if the environment contains an obstacle that could be circumnavigated in several different ways, and the policies in E_i and E_j both need to circumnavigate it before doing something else. This means that $\dim(Z_1 \cup \dots \cup Z_m)$ is large when, roughly, either (i) we have very large and diverse sets of policies in $\hat{\Pi}$ that get the same reward according to $\mathcal{R}$, or (ii) we have a large number of different sets of policies that get the same reward according to $\mathcal{R}$, and where there are different kinds of diversity in the behaviour of the policies in each set. There are also intuitive special cases of Theorem 3. For example, as noted before, if E_i is a singleton then Z_i has no impact on $\dim(Z_1 \cup \dots \cup Z_m)$. This implies the following corollary:

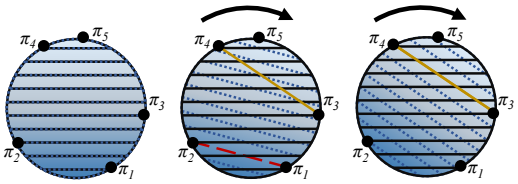


Figure 4: An illustration of the state-action occupancy space with a reward function defined over it. Points correspond to policies' state-action occupancies. Shading intensity indicates expected reward. Rotating the reward function to make $J(\pi_3) > J(\pi_4)$ passes through a reward which sets $J(\pi_1) = J(\pi_2)$. Solid black lines are contour lines of the original reward function, dotted blue lines are contour lines of the rotated reward function.

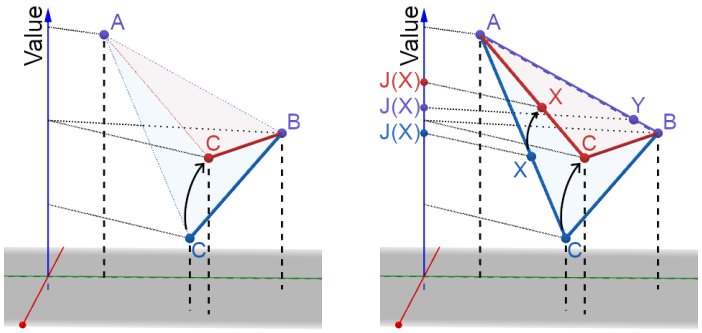
Corollary 3. *For any finite set of policies $\hat{\Pi}$, any environment, and any reward function $\mathcal{R}$, if $|\hat{\Pi}| \geq 2$ and $J(\pi) \neq J(\pi')$ for all $\pi, \pi' \in \hat{\Pi}$ then there is a non-trivial simplification of $\mathcal{R}$.*

A natural question is whether there always is a simplification on the set of all deterministic policies. As it turns out, this is not the case. For concreteness, and to build intuition for this result, we examine the set of deterministic policies in a simple $MDP \setminus \mathcal{R}$ with $S = \{0, 1\}, A = \{0, 1\}, T(s, a) = a, I = \{0 : 0.5, 1 : 0.5\}, \gamma = 0.5$. Denote π_{ij} the policy that takes action i from state 0 and action j from state 1. There are exactly four deterministic policies. We find that of the $4! = 24$ possible policy orderings, 12 are realizable via some reward function. In each of those 12 orderings, exactly two policies (of the six available pairs of policies in the ordering) can be set to equal value without resulting in the trivial reward function (*which* pair can be equated depends on the ordering in consideration). Attempting to set three policies equal always results in the trivial reward simplification.

For example, given the ordering $\pi_{00} \leq \pi_{01} \leq \pi_{11} \leq \pi_{10}$, we can achieve $\pi_{00} = \pi_{01}$ and make the other inequalities strict by setting the rewards to $r = [[0, 3], [2, 1]]$. But for this ordering, there is no reward assignment other than the trivial one that achieves $\pi_{01} = \pi_{11}$ or $\pi_{11} = \pi_{10}$ while respecting the other inequalities. We develop and release a software suite to compute these results. Given an environment and a set of policies, it can calculate all policy orderings representable by some reward function. Also, for a given policy ordering, it can calculate all nontrivial simplifications and reward functions that represent them. For a link to the repository, as well as a full exploration of these policies, orderings, and simplifications, see Appendix. For a link to the repository, see Appendix.

5.3 Ungameability in Infinite Policy Sets

The results in Section 5.1 do not characterize ungameability for infinite policy sets that do not contain open sets. Here, we provide two examples of such policy sets; one of them admits ungameable reward pairs and the other does not. Consider policies A, B, C, and reward functions $\mathcal{R}_1$ with $J_1(C) < J_1(B) < J_1(A)$ and $\mathcal{R}_2$ with $J_2(C) = J_2(B) < J_2(A)$. Policy sets $\Pi_a = \{A\} \cup \{\lambda B + (1-\lambda)C : \lambda \in [0,1]\}$ and $\Pi_b = \{A\} \cup \{\lambda B + (1-\lambda)C : \lambda \in [0,1]\}$ are depicted in Figure 5; the vertical axis represents policies' values according to $\mathcal{R}_1$ and $\mathcal{R}_2$. For Π_a, $\mathcal{R}_2$ is a simplification of $\mathcal{R}_1$, but for Π_b, it is not, since $J_1(X) < J_1(Y)$ and $J_2(X) > J_2(Y)$.



(a) (b)

Figure 5: Points A, B, C represent deterministic policies, while the bold lines between them represent stochastic policies. In (a), we only allow mixtures of B and C, while in (b), we allow mixtures of any two policies. Value of a policy according to a given reward function can be visualized by projecting the policy onto the vertical axis. We attempt to simplify the reward function by rotating the blue triangle along the AB axis, leading to the red triangle. In (a), when we don't allow mixtures involving A, the simplification is successful, and sets all policies along the BC segment equal. In (b), such a simplification is not possible, since it would swap the relative value of policies X and Y. Indeed, in (b), only the trivial simplification is possible; all others result in gameable reward functions.

6 Discussion

We reflect on our results and identify limitations in Section 6.1. We discuss how our work can inform discussions about the appropriateness, potential risks, and limitations of using of reward functions as specifications in Section 6.2.

6.1 Limitations

Our work has a number of limitations. We have only considered finite MDPs and Markov reward functions, leaving more general environments for future work. While we characterized gameability and simplification for finite policy sets, the conditions for simplification are somewhat opaque, and our characterization of infinite policy sets remains incomplete.

As previously discussed, our definition of gameability is strict, arguably too strict. Nonetheless, we believe that understanding the consequences of this strict definition is an important starting point for further theoretical work in this area.

The main issue with the strictness of our definition has to do with the symmetric nature of gameability. The existence of complex behaviors that yield low proxy reward and high true reward is much less concerning than the reverse, as these behaviors are unlikely to be discovered while optimizing the proxy. For example, it is very unlikely that our agent would solve climate change in the course of learning how to wash dishes. Note that the existence of *simple* behaviors with low proxy reward and high true reward *is* concerning; these could arise early in training, leading us to trust the proxy, only to later see the true reward decrease as the proxy is further optimized. To account for this issue, future work should explore realistic assumptions about the probability of encountering a given sequence of policies when optimizing the proxy, and measure gameability in proportion to this probability.

We could allow for approximate ungameability by only considering pairs of policies ranked differently by the true and proxy reward functions as evidence of gaming iff their value according to the true reward function differs by more than some ε. Probabilistic ungameability could be defined by looking at the number of misordered policies; this would seem to require making assumptions about the probability of encountering a given policy when optimizing the proxy.

Finally, while our work theoretically characterizes the gameability relationship, gameability is far from a guarantee of gaming. Extensive empirical work is necessary to better understand the factors that influence the occurrence and severity of reward gaming in practice.

6.2 Implications

How should we specify our preferences for AI systems' behavior? And how detailed a specification is required to achieve a good outcome? In reinforcement learning, the goal of maximizing (some) reward function is often taken for granted, but a number of scholars have expressed reservations about this approach (Dobbe et al., 2021; Hadfield-Menell et al., 2016b, 2017; Bostrom, 2014). Our work has several implications for this discussion, although we caution against drawing any strong conclusions due to the limitations mentioned in Section 6.1.

One source of confusion and disagreement is the role of the reward function; it is variously considered as a means of specifying a task (Leike et al., 2018) or encoding broad human values (Dewey, 2011). We might hope to use Markov reward functions to specify narrow tasks without risking behavior that goes against our broad values. However, if we consider the "narrow task" reward function as a proxy for the true "broad values" reward function, our results indicate that this is not possible: these two reward functions will invariably be gameable. Such reasoning suggests that reward functions must instead encode broad human values. This seems challenging, perhaps intractably so, indicating that alternatives to reward optimization may be more promising. Potential alternatives include imitation learning (Ross et al., 2011), constrained RL (Szepesvári, 2020), quantilizers (Taylor, 2016), and incentive management (Everitt et al., 2019).

Scholars have also criticized the assumption that human values can be encoded as rewards (Dobbe et al., 2021), and challenged the use of metrics more broadly (O'Neil, 2016; Thomas and Uminsky, 2022), citing Goodhart's Law (Manheim and Garrabrant, 2018; Goodhart, 1975). A concern more specific to the optimization of reward functions is power-seeking (Turner et al., 2021; Bostrom, 2012; Omohundro, 2008). Turner et al. (2021) prove that optimal policies tend to seek power in most MDPs and for most reward functions. Such behavior could lead to human disempowerment; for instance, an AI system might disable its off-switch (Hadfield-Menell et al., 2016a). Bostrom (2014) and others have argued that power-seeking makes even slight misspecification of rewards potentially catastrophic.

Despite such concerns, approaches to specification based on learning reward functions remain popular (Fu et al., 2017; Stiennon et al., 2020; Nakano et al., 2021). So far, reward gaming has usually been avoidable in practice, although some care must be taken (Stiennon et al., 2020). Proponents of such approaches have emphasized the importance of learning a reward model in order to exceed human performance and generalize to new settings (Brown et al., 2020a; Leike et al., 2018). But our work indicates that such learned rewards are almost certainly gameable, and so cannot be safely optimized. Thus we recommend viewing such approaches as a means of learning a policy in a controlled setting, which should then be validated before being deployed.

7 Conclusion

Our work begins the formal study of reward gaming in reinforcement learning. We formally define gameability and simplification of reward functions, and show conditions for the (non-)existence of non-trivial examples of each. We find that ungameability is quite a strict condition, as the set of all policies never contains non-trivial ungameable pairs of reward functions. Thus in practice, reward gaming must be prevented by limiting the set of possible policies, or controlling (e.g., limiting) optimization. Alternatively, we could pursue approaches not based on optimizing reward functions.

References

Abel, D., Dabney, W., Harutyunyan, A., Ho, M. K., Littman, M., Precup, D., and Singh, S. (2021). On the Expressivity of Markov Reward. *Advances in Neural Information Processing Systems*, 34.

Amodei, D., Christiano, P., and Ray, A. (2017). Learning from Human Preferences. OpenAI `https://openai.com/blog/deep-reinforcement-learning-from-human-preferences/`.

Bird, J. and Layzell, P. (2002). The evolved radio and its implications for modelling the evolution of novel sensors. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 2, pages 1836–1841. IEEE.

Bostrom, N. (2012). The superintelligent will: Motivation and instrumental rationality in advanced artificial agents. *Minds and Machines*, 22(2):71–85.

Bostrom, N. (2014). Superintelligence: Paths, Dangers, Strategies.

Brown, D. S., Goo, W., and Niekum, S. (2020a). Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on robot learning*, pages 330–359. PMLR.

Brown, D. S., Schneider, J., Dragan, A. D., and Niekum, S. (2020b). Value Alignment Verification. *CoRR*, abs/2012.01557.

Clark, J. and Amodei, D. (2016). Faulty Reward Functions in the Wild. OpenAI Codex `https://openai.com/blog/faulty-reward-functions/`.

Dewey, D. (2011). Learning What to Value. In Schmidhuber, J., Thórisson, K. R., and Looks, M., editors, *Artificial General Intelligence: 4th International Conference, AGI 2011*, pages 309–314, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dobbe, R., Gilbert, T. K., and Mintz, Y. (2021). Hard Choices in Artificial Intelligence. *CoRR*, abs/2106.11022.

Everitt, T., Ortega, P. A., Barnes, E., and Legg, S. (2019). Understanding agent incentives using causal influence diagrams. Part I: Single action settings. *arXiv preprint arXiv:1902.09980*.

Fu, J., Luo, K., and Levine, S. (2017). Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*.

Golden, D. (2001). Glass Floor: Colleges Reject Top Applicants, Accepting Only the Students Likely to Enroll. *The Wall Street Journal*. `https://www.wsj.com/articles/SB991083160294634500`.

Goodhart, C. A. (1975). Problems of monetary management: the UK experience. In of Australia, R. B., editor, *Papers in monetary economics*. Reserve Bank of Australia.

Hadfield-Menell, D., Dragan, A., Abbeel, P., and Russell, S. (2016a). The Off-Switch Game. *CoRR*, abs/1611.08219.

Hadfield-Menell, D., Dragan, A. D., Abbeel, P., and Russell, S. J. (2016b). Cooperative Inverse Reinforcement Learning. *CoRR*, abs/1606.03137.

Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S. J., and Dragan, A. (2017). Inverse reward design. *Advances in neural information processing systems*, 30.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. (2018). Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31.

Krakovna, V., Orseau, L., Kumar, R., Martic, M., and Legg, S. (2018). Penalizing side effects using stepwise relative reachability. *CoRR*, abs/1806.01186.

Krakovna, V., Uesato, J., Mikulik, V., Rahtz, M., Everitt, T., Kumar, R., Kenton, Z., Leike, J., and Legg, S. (2020). Specification gaming: the flip side of AI ingenuity.

Leike, J., Krueger, D., Everitt, T., Martic, M., Maini, V., and Legg, S. (2018). Scalable agent alignment via reward modeling: a research direction. *CoRR*, abs/1811.07871.

Manheim, D. and Garrabrant, S. (2018). Categorizing Variants of Goodhart's Law. *CoRR*, abs/1803.04585.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. (2021). WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287.

Ng, A. Y., Russell, S. J., et al. (2000). Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2.

Omohundro, S. M. (2008). The basic AI drives.

O'Neil, C. (2016). *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown Publishing Group.

Pan, A., Bhatia, K., and Steinhardt, J. (2022). The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models. *arXiv preprint arXiv:2201.03544*.

Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Szepesvári, C. (2020). Constrained MDPs and the reward hypothesis. `http://readingsml.blogspot.com/2020/03/constrained-mdps-and-reward-hypothesis.html`.

Taylor, J. (2016). Quantilizers: A safer alternative to maximizers for limited optimization. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*.

Thomas, R. L. and Uminsky, D. (2022). Reliance on metrics is a fundamental challenge for AI. *Patterns*, 3(5):100476.

Thrun, S. and Schwartz, A. (1993). Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 Connectionist Models Summer School Hillsdale, NJ. Lawrence Erlbaum*, volume 6.

Turner, A. M., Hadfield-Menell, D., and Tadepalli, P. (2019). Conservative Agency via Attainable Utility Preservation. *CoRR*, abs/1902.09725.

Turner, A. M., Smith, L., Shah, R., Critch, A., and Tadepalli, P. (2021). Optimal Policies Tend to Seek Power. *Advances in Neural Information Processing Systems*.

Zhang, B., Rajan, R., Pineda, L., Lambert, N., Biedenkapp, A., Chua, K., Hutter, F., and Calandra, R. (2021). On the Importance of Hyperparameter Optimization for Model-based Reinforcement Learning. *CoRR*, abs/2102.13651.

Zhuang, S. and Hadfield-Menell, D. (2020). Consequences of misaligned AI. *Advances in Neural Information Processing Systems*, 33:15763–15773.

Ziebart, B. D. (2010). *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.2
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Some of the proofs are in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code and instructions for running it are available in the supplementary materials. The code does not use any datasets.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] We do not perform model training in this work.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] No compute beyond a personal laptop (with integrated graphics) was used.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A] The codebase was written from scratch.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The codebase is available in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Overview

Section B contains proofs of the main theoretical results. Section C expands on examples given in the main text. Section D presents an ungameability diagram for a generic set of three policies a, b, c; Section E shows a simplification diagram of the same policies.

B Proofs

Before proving our results, we restate assumptions and definitions. First, recall the preliminaries from Section 4.1, and in particular, that we use $\mathcal{F} : \Pi \to \mathbb{R}^{|S||A|}$ to denote the embedding of policies into Euclidean space via their discounted state-action visit counts, i.e.;

$$\mathcal{F}(\pi)[s, a] = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(S_t = s, A_t = a).$$

Given a reward function $\mathcal{R}$, let $\vec{\mathcal{R}} \in \mathbb{R}^{|S||A|}$ be the vector where $\vec{\mathcal{R}}[s, a] = \mathbb{E}_{S' \sim T(s,a)}[\mathcal{R}(s, a, S')]$. Note that $J(\pi) = \mathcal{F}(\pi) \cdot \vec{\mathcal{R}}$.

We say $\mathcal{R}_1$ and $\mathcal{R}_2$ are **equivalent** on a set of policies Π if J_1 and J_2 induce the same ordering of Π, and that $\mathcal{R}$ is **trivial** on Π if $J(\pi) = J(\pi')$ for all $\pi, \pi' \in \Pi$. We also have the following definitions from Sections 4 and 5:

Definition 1. A pair of reward functions $\mathcal{R}_1$, $\mathcal{R}_2$ are **gameable** relative to policy set Π and an environment $(S, A, T, I, __, \gamma)$ if there exist $\pi, \pi' \in \Pi$ such that

$$J_1(\pi) < J_1(\pi') \;\;\&\;\; J_2(\pi) > J_2(\pi'),$$

else they are **ungameable**.

Definition 2. $\mathcal{R}_2$ is a **simplification** of $\mathcal{R}_1$ relative to policy set Π if for all $\pi, \pi' \in \Pi$,

$$J_1(\pi) < J_1(\pi') \implies J_2(\pi) \leq J_2(\pi') \;\;\&\;\; J_1(\pi) = J_1(\pi') \implies J_2(\pi) = J_2(\pi')$$

and there exist $\pi, \pi' \in \Pi$ such that $J_2(\pi) = J_2(\pi')$ but $J_1(\pi) \neq J_1(\pi')$. Moreover, if $\mathcal{R}_2$ is trivial then we say that this is a **trivial simplification**.

Note that these definitions only depend on the policy orderings associated with $\mathcal{R}_2$ and $\mathcal{R}_1$, and so we can (and do) also speak of (ordered) pairs of policy orderings being simplifications or gameable. We also make use of the following definitions:

Definition 3. A (stationary) policy π is ε-suboptimal if $J(\pi) \geq J(\pi^\star) - \varepsilon$, where $\varepsilon > 0$

Definition 4. A (stationary) policy π is δ-deterministic if $\forall s \in S \; \exists a \in A : \mathbb{P}(\pi(s) = a) \geq \delta$, where $\delta < 1$.

B.1 Non-trivial Ungameability Requires Restricting the Policy Set

Formally, a set of (stationary) policies $\dot{\Pi}$ is **open** if $\mathcal{F}(\dot{\Pi})$ is open in the smallest affine space that contains all stationary policies (also represented as $|S||A|$-dimensional vectors). This space is $|S|(|A| - 1)$-dimensional, since all action probabilities sum to 1.

We require two more propositions for the proof of this lemma.

Proposition 1. *If $\dot{\Pi}$ is open then $\mathcal{F}$ is injective on $\dot{\Pi}$.*

Proof. First note that, since $\pi(a \mid s) \geq 0$, we have that if $\dot{\Pi}$ is open then $\pi(a \mid s) > 0$ for all s, a for all $\pi \in \dot{\Pi}$. In other words, all policies in $\dot{\Pi}$ take each action with positive probability in each state.

Now suppose $\mathcal{F}(\pi) = \mathcal{F}(\pi')$ for some $\pi, \pi' \in \tilde{\Pi}$. Next, define w_π as

$$w_\pi(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_{\tau \sim \pi}(S_t = s).$$

Note that if $\mathcal{F}(\pi) = \mathcal{F}(\pi')$ then $w_\pi = w_{\pi'}$, and moreover that

$$\mathcal{F}(\pi)[s, a] = w_\pi(s)\pi(a \mid s).$$

Next, since π takes each action with positive probability in each state, we have that π visits every state with positive probability. This implies that $w_\pi(s) \neq 0$ for all s, which means that we can express π as

$$\pi(a \mid s) = \frac{\mathcal{F}(\pi)[s, a]}{w_\pi(s)}.$$

This means that if $\mathcal{F}(\pi) = \mathcal{F}(\pi')$ for some $\pi, \pi' \in \tilde{\Pi}$ then $\pi = \pi'$. □

Note that $\mathcal{F}$ is *not* injective on Π; if there is some state s that π reaches with probability 0, then we can alter the behaviour of π at s without changing $\mathcal{F}(\pi)$. But every policy in an open policy set $\dot{\Pi}$ visits every state with positive probability, which then makes $\mathcal{F}$ injective. In fact, Proposition 1 straightforwardly generalises to the set of all policies that visit all states with positive probability.

Proposition 2. $\mathrm{Im}(\mathcal{F})$ *is located in a linear subspace with* $|S|(|A| - 1)$ *dimensions.*

Proof. To show that $\mathrm{Im}(\mathcal{F})$ is located in a linear subspace with $|S|(|A| - 1)$ dimensions, first note that

$$\sum_{s,a} \mathcal{F}(\pi)[s, a] = \sum_{t=0}^{\infty} \gamma^t = \frac{1}{1 - \gamma}$$

for all π. That is, $\mathrm{Im}(\mathcal{F})$ is located in an affine space of points with a fixed ℓ_1-norm, and this space does not contain the origin.

Next, note that $J(\pi) = \mathcal{F}(\pi) \cdot \vec{\mathcal{R}}$. This means that if knowing the value of J for all π determines $\vec{\mathcal{R}}$ modulo at least n free variables, then $\mathrm{Im}(\mathcal{F})$ contains at most $|S||A| - n$ linearly independent vectors. Next recall *potential shaping* (Ng et al., 1999). In brief, given a reward function $\mathcal{R}$ and a *potential function* $\Phi : S \to \mathbb{R}$, we can define a *shaped reward function* $\mathcal{R}'$ by

$$\mathcal{R}'(s, a, s') = \mathcal{R}(s, a, s') + \gamma\Phi(s') - \Phi(s),$$

or, alternatively, if we wish $\mathcal{R}'$ to be defined over the domain $S \times A$,

$$\mathcal{R}'(s, a) = \mathcal{R}(s, a) + \gamma\mathbb{E}_{S' \sim T(s,a)}[\Phi(S')] - \Phi(s).$$

In either case, it is possible to show that if $\mathcal{R}'$ is produced by shaping $\mathcal{R}$ with Φ, and $\mathbb{E}_{S_0 \sim I}[\Phi(S_0)] = 0$, then $J(\pi) = J'(\pi)$ for all π. This means that knowing the value of $J(\pi)$ for all π determines $\vec{\mathcal{R}}$ modulo at least $|S| - 1$ free variables, which means that $\mathrm{Im}(\mathcal{F})$ contains at most $|S||A| - (|S| - 1) = |S|(|A| - 1) + 1$ linearly independent vectors. Since the smallest affine space that contains $\mathrm{Im}(\mathcal{F})$ does *not* contain the origin, this in turn means that $\mathrm{Im}(\mathcal{F})$ is located in a linear subspace with $= |S|(|A| - 1) + 1 - 1 = |S|(|A| - 1)$ dimensions. □

Lemma 1. *In any* $MDP \setminus \mathcal{R}$*, if* $\dot{\Pi}$ *is an open set of policies, then* $\mathcal{F}(\dot{\Pi})$ *is open in* $\mathbb{R}^{|S|(|A|-1)}$*, and* $\mathcal{F}$ *is a homeomorphism between* $\dot{\Pi}$ *and* $\mathcal{F}(\dot{\Pi})$*.*

Proof. By the Invariance of Domain Theorem, if

1. U is an open subset of $\mathbb{R}^n$, and
2. $f : U \to \mathbb{R}^n$ is an injective continuous map,

then $f(U)$ is open in $\mathbb{R}^n$ and f is a homeomorphism between U and $f(U)$. We will show that $\mathcal{F}$ and $\dot{\Pi}$ satisfy the requirements of this theorem.

We begin by noting that Π can be represented as a set of points in $\mathbb{R}^{|S|(|A|-1)}$. We do this by considering each policy π as a vector $\vec{\pi}$ of length $|S||A|$, where $\vec{\pi}[s, a] = \pi(a \mid s)$. We also have $\sum_{a \in A} \pi(a \mid s) = 1$ for all s, which means that once we have set the probabilities of $\pi(a \mid s)$ for each $a \in A \setminus \{a'\}$, then $\pi(a' \mid s)$ is also determined; this removes one degree of freedom for each state. From now on, we will assume that Π is embedded in $\mathbb{R}^{|S|(|A|-1)}$ in this way.

By assumption, $\dot{\Pi}$ is an open set in $\mathbb{R}^{|S|(|A|-1)}$. Moreover, by Proposition 2, we have that $\mathcal{F}$ is (isomorphic to) a mapping $\dot{\Pi} \to \mathbb{R}^{|S|(|A|-1)}$. By Proposition 1, we have that $\mathcal{F}$ is injective on $\dot{\Pi}$. Finally, $\mathcal{F}$ is continuous; this can be seen from its definition. We can therefore apply the Invariance of Domain Theorem, and obtain that $\mathcal{F}(\dot{\Pi})$ is open in $\mathbb{R}^{|S|(|A|-1)}$, and that $\mathcal{F}$ is a homeomorphism between $\dot{\Pi}$ and $\mathcal{F}(\dot{\Pi})$. □

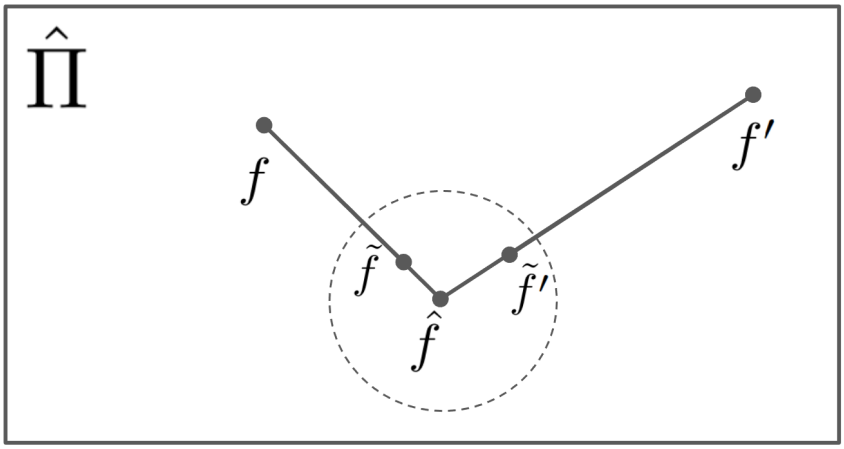


Figure 6: Illustration of the various realizable feature counts used in the proof of Theorem 1.

Theorem 1. *In any $MDP \setminus \mathcal{R}$, if $\hat{\Pi}$ contains an open set, then any pair of reward functions that are ungameable and non-trivial on $\hat{\Pi}$ are equivalent on $\hat{\Pi}$.*

Proof. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be any two ungameable and non-trivial reward functions. We will show that, for any $\pi, \pi' \in \hat{\Pi}$, we have $J_1(\pi) = J_1(\pi') \implies J_2(\pi) = J_2(\pi')$, and thus, by symmetry, $J_1(\pi) = J_1(\pi') \iff J_2(\pi) = J_2(\pi')$. Since $\mathcal{R}_1$ and $\mathcal{R}_2$ are ungameable, this further means that they have exactly the same policy order, i.e. that they are equivalent.

Choose two arbitrary $\pi, \pi' \in \hat{\Pi}$ with $J_1(\pi) = J_1(\pi')$ and let $f \doteq \mathcal{F}(\pi)$, $f' \doteq \mathcal{F}(\pi')$. The proof has 3 steps:

1. We find analogues for f and f', $\tilde{f}$ and $\tilde{f}'$, within the same open ball in $\mathcal{F}(\hat{\Pi})$.
2. We show that the tangent hyperplanes of $\vec{R}_1$ and $\vec{R}_2$ at $\tilde{f}$ must be equal to prevent neighbors of $\tilde{f}$ from making $\mathcal{R}_1$ and $\mathcal{R}_2$ gameable.
3. We use linearity to show that this implies that $J_2(\pi) = J_2(\pi')$.

Step 1: By assumption, $\hat{\Pi}$ contains an open set $\dot{\Pi}$. Let $\hat{\pi}$ be some policy in $\dot{\Pi}$, and let $\hat{f} \doteq \mathcal{F}(\hat{\pi})$. Since $\dot{\Pi}$ is open, Lemma 1 implies that $\mathcal{F}(\dot{\Pi})$ is open in $\mathbb{R}^{|S|(|A|-1)}$. This means that, if v, v' are the vectors such that $\hat{f} + v = f$ and $\hat{f} + v' = f'$, then there is a positive but sufficiently small δ such that $\tilde{f} \doteq \hat{f} + \delta v$ and $\tilde{f}' \doteq \hat{f} + \delta v'$ both are located in $\mathcal{F}(\dot{\Pi})$, see Figure 6. This further implies that there are policies $\tilde{\pi}, \tilde{\pi}' \in \dot{\Pi}$ such that $\mathcal{F}(\tilde{\pi}) = \tilde{f}$ and $\mathcal{F}(\tilde{\pi}') = \tilde{f}'$.

Step 2: Recall that $J(\pi) = \mathcal{F}(\pi) \cdot \vec{\mathcal{R}}$. Since $\mathcal{R}_1$ is non-trivial on $\hat{\Pi}$, it induces a $(|S|(|A|-1)-1)$-dimensional hyperplane tangent to $\vec{\mathcal{R}}_1$ corresponding to all points $x \in \mathbb{R}^{|S|(|A|-1)}$ such that $x \cdot \vec{\mathcal{R}}_1 = \tilde{f} \cdot \vec{\mathcal{R}}_1$, and similarly for $\mathcal{R}_2$. Call these hyperplanes H_1 and H_2, respectively. Note that $\tilde{f}$ is contained in both H_1 and H_2.

Next suppose $H_1 \neq H_2$. Then, we would be able to find a point $f_{12} \in \mathcal{F}(\dot{\Pi})$, such that $f_{12} \cdot \vec{\mathcal{R}}_1 > \tilde{f} \cdot \vec{\mathcal{R}}_1$ but $f_{12} \cdot \vec{\mathcal{R}}_2 < \tilde{f} \cdot \vec{\mathcal{R}}_2$. This, in turn, means that there is a policy $\pi_{12} \in \dot{\Pi}$ such that $\mathcal{F}(\pi_{12}) = f_{12}$, and such that $J_1(\pi_{12}) > J_1(\tilde{\pi})$ but $J_2(\pi_{12}) < J_2(\tilde{\pi})$. Since $\mathcal{R}_1$ and $\mathcal{R}_2$ are ungameable, this is a contradiction. Thus $H_1 = H_2$.

Step 3: Since $J_1(\pi) = J_1(\pi')$, we have that $f \cdot \vec{\mathcal{R}}_1 = f' \cdot \vec{\mathcal{R}}_1$. By linearity, this implies that $\tilde{f} \cdot \vec{\mathcal{R}}_1 = \tilde{f}' \cdot \vec{\mathcal{R}}_1$; we can see this by expanding $\tilde{f} = \hat{f} + \delta v$ and $\tilde{f}' = \hat{f} + \delta v'$. This means that $\tilde{f}' \in H_1$. Now, since $H_1 = H_2$, this means that $\tilde{f}' \in H_2$, which in turn implies that $\tilde{f} \cdot \vec{\mathcal{R}}_2 = \tilde{f}' \cdot \vec{\mathcal{R}}_2$. By linearity, this then further implies that $f \cdot \vec{\mathcal{R}}_2 = f' \cdot \vec{\mathcal{R}}_2$, and hence that $J_2(\pi) = J_2(\pi')$. Since π, π' were chosen arbitrarily, this means that $J_1(\pi) = J_1(\pi') \implies J_2(\pi) = J_2(\pi')$. □

Corollary 1. *In any $MDP \setminus \mathcal{R}$, any pair of reward functions that are ungameable and non-trivial on the set of all (stationary) policies Π are equivalent on Π.*

Proof. This corollary follows from Theorem 1, if we note that the set of all policies does contain an open set. This includes, for example, the set of all policies in an ϵ-ball around the policy that takes all actions with equal probability in each state. □

Corollary 2. *In any $MDP \setminus \mathcal{R}$, any pair of reward functions that are ungameable and non-trivial on the set of all ε-suboptimal policies ($\varepsilon > 0$) Π^ε are equivalent on Π^ε, and any pair of reward functions that are ungameable and non-trivial on the set of all δ-deterministic policies ($\delta < 1$) Π^δ are equivalent on Π^δ.*

Proof. To prove this, we will establish that both Π^ε and Π^δ contain open policy sets, and then apply Theorem 1.

Let us begin with Π^δ. First, let π be some deterministic policy, and let π_ϵ be the policy that in each state with probability $1 - \epsilon$ takes the same action as π, and otherwise samples an action uniformly. Then if $\delta < \epsilon < 1$, π_ϵ is the center of an open ball in Π^δ. Thus Π^δ contains an open set, and we can apply Theorem 1.

For Π^ε, let $\pi^\star$ be an optimal policy, and apply an analogous argument. □

B.2 Finite Policy Sets

Theorem 2. *For any $MDP \setminus \mathcal{R}$, any finite set of policies $\hat{\Pi}$ containing at least two π, π' such that $\mathcal{F}(\pi) \neq \mathcal{F}(\pi')$, and any reward function $\mathcal{R}_1$, there is a non-trivial reward function $\mathcal{R}_2$ such that $\mathcal{R}_1$ and $\mathcal{R}_2$ are ungameable but not equivalent.*

Proof. If $\mathcal{R}_1$ is trivial, then simply choose any non-trivial $\mathcal{R}_2$. Otherwise, the proof proceeds by finding a path from $\vec{\mathcal{R}}_1$ to $-\vec{\mathcal{R}}_1$, and showing that there must be an $\vec{\mathcal{R}}_2$ on this path such that $\mathcal{R}_2$ is non-trivial and ungameable with respect to $\mathcal{R}_1$, but not equivalent to $\mathcal{R}_1$.

The key technical difficulty is to show that there exists a continuous path from $\mathcal{R}_1$ to $-\mathcal{R}_1$ in $\mathbb{R}^{|S||A|}$ that does not include any trivial reward functions. Once we've established that, we can simply look for the first place where an inequality is reversed – because of continuity, it first becomes an equality. We call the reward function at that point $\mathcal{R}_2$, and note that $\mathcal{R}_2$ is ungameable wrt $\mathcal{R}_1$ and not equivalent to $\mathcal{R}_1$. We now walk through the technical details of these steps.

First, note that $J(\pi) = \mathcal{F}(\pi) \cdot \vec{\mathcal{R}}$ is continuous in $\vec{\mathcal{R}}$. This means that if $J_1(\pi) > J_2(\pi')$ then there is a unique first vector $\vec{\mathcal{R}}_2$ on any path from $\vec{\mathcal{R}}_1$ to $-\vec{\mathcal{R}}_1$ such that $\mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_2 \not> \mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_2$, and for this vector we have that $\mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_2 = \mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_2$. Since $\hat{\Pi}$ is finite, and since $\mathcal{R}_1$ is not trivial, this means that on any path from $\vec{\mathcal{R}}_1$ to $-\vec{\mathcal{R}}_1$ there is a unique first vector $\vec{\mathcal{R}}_2$ such that $\mathcal{R}_2$ is not equivalent to $\mathcal{R}_1$, and then $\mathcal{R}_2$ must also be a ungameable with respect to $\mathcal{R}_1$.

It remains to show that there is a path from $\vec{\mathcal{R}}_1$ to $-\vec{\mathcal{R}}_1$ such that no vector along this path corresponds to a trivial reward function. Once we have such a path, the argument above implies that $\mathcal{R}_2$ must be a non-trivial reward function that is ungameable with respect to $\mathcal{R}_1$. We do this using a dimensionality argument. If $\mathcal{R}$ is trivial on $\hat{\Pi}$, then there is some $c \in \mathbb{R}$ such that $\mathcal{F}(\pi) \cdot \vec{\mathcal{R}} = c$ for all $\pi \in \hat{\Pi}$. This means that if $\mathcal{F}(\hat{\Pi})$ has at least d linearly independent vectors, then the set of all such vectors $\vec{\mathcal{R}}$ forms a linear subspace with at most $|S||A| - d$ dimensions. Now, since $\hat{\Pi}$ contains at least two π, π' such that $\mathcal{F}(\pi) \neq \mathcal{F}(\pi')$, we have that $\mathcal{F}(\hat{\Pi})$ has at least 2 linearly independent vectors, and hence that the set of all reward functions that are trivial on $\hat{\Pi}$ forms a linear subspace with at most $|S||A| - 2$ dimensions. This means that there must exist a path from $\vec{\mathcal{R}}_1$ to $-\vec{\mathcal{R}}_1$ that avoids this

subspace, since only a hyperplane (with dimension $|S||A|-1$) can split $\mathbb{R}^{|S||A|}$ into two disconnected components. $\square$

Theorem 3. *Let $\hat{\Pi}$ be a finite set of policies, and $\mathcal{R}$ a reward function. The following procedure determines if there exists a non-trivial simplification of $\mathcal{R}$ in a given $MDP \setminus \mathcal{R}$:*

1. *Let $E_1 \dots E_m$ be the partition of $\hat{\Pi}$ where π, π' belong to the same set iff $J(\pi) = J(\pi')$.*
2. *For each such set E_i, select a policy $\pi_i \in E_i$ and let Z_i be the set of vectors that is obtained by subtracting $\mathcal{F}(\pi_i)$ from each element of $\mathcal{F}(E_i)$.*

Then there is a non-trivial simplification of $\mathcal{R}$ iff $\dim(Z_1 \cup \dots \cup Z_m) \leq \dim(\mathcal{F}(\hat{\Pi})) - 2$, where $\dim(S)$ is the number of linearly independent vectors in S.

Proof. This proof uses a similar proof strategy as Theorem 2. However, in addition to avoiding trivial reward functions on the path from $\vec{\mathcal{R}}_1$ to $-\vec{\mathcal{R}}_1$, we must also ensure that we stay within the "equality-preserving space", to be defined below.

First recall that $\mathcal{F}(\hat{\Pi})$ is a set of vectors in $\mathbb{R}^{|S||A|}$. If $\dim(\mathcal{F}(\hat{\Pi})) = D$ then these vectors are located in a D-dimensional linear subspace. Therefore, we will consider $\mathcal{F}(\hat{\Pi})$ to be a set of vectors in $\mathbb{R}^D$. Next, recall that any reward function $\mathcal{R}$ induces a linear function L on $\mathbb{R}^D$, such that $J = L \circ \mathcal{F}$, and note that there is a D-dimensional vector $\vec{\mathcal{R}}$ that determines the *ordering* that $\mathcal{R}$ induces over all points in $\mathbb{R}^D$. To determine the *values* of J on all points in $\mathbb{R}^D$ we would need a $(D+1)$-dimensional vector, but to determine the *ordering*, we can ignore the height of the function. In other words, $L(x) = x \cdot \vec{\mathcal{R}} + L(\vec{0})$, for any $x \in \mathbb{R}^D$. Note that this is a different vector representation of reward functions than that which was used in Theorem 2 and before.

Suppose $\mathcal{R}_2$ is a reward function such that if $J_1(\pi) = J_1(\pi')$ then $J_2(\pi) = J_2(\pi')$, for all $\pi, \pi' \in \hat{\Pi}$. This is equivalent to saying that $L_2(\mathcal{F}(\pi)) = L_2(\mathcal{F}(\pi'))$ if $\pi, \pi' \in E_i$ for some E_i. By the properties of linear functions, this implies that if $\mathcal{F}(E_i)$ contains d_i linearly independent vectors then it specifies a $(d_i - 1)$-dimensional affine space S_i such that $L_2(x) = L_2(x')$ for all points $x, x' \in S_i$. Note that this is the smallest affine space which contains all points in E_i. Moreover, L_2 is also constant for any affine space $\bar{S}_i$ *parallel* to S_i. Formally, we say that $\bar{S}_i$ is parallel to S_i if there is a vector z such that for any $y \in \bar{S}_i$ there is an $x \in S_i$ such that $y = x + z$. From the properties of linear functions, if $L_2(x) = L_2(x')$ then $L_2(x + z) = L_2(x' + z)$.

Next, from the transitivity of equality, if we have two affine spaces $\bar{S}_i$ and $\bar{S}_j$, such that L_2 is constant over each of $\bar{S}_i$ and $\bar{S}_j$, and such that $\bar{S}_i$ and $\bar{S}_j$ *intersect*, then L_2 is constant over all points in $\bar{S}_i \cup \bar{S}_j$. From the properties of linear functions, this then implies that L_2 is constant over all points in the smallest affine space $\bar{S}_i \otimes \bar{S}_j$ containing $\bar{S}_i$ and $\bar{S}_j$, given by combining the linearly independent vectors in $\bar{S}_i$ and $\bar{S}_j$. Note that $\bar{S}_i \otimes \bar{S}_j$ has between $\max(d_i, d_j)$ and $(d_i + d_j - 1)$ dimensions. In particular, since the affine spaces of $Z_1 \dots Z_m$ intersect (at the origin), and since L_2 is constant over these spaces, we have that L_2 must be constant for all points in the affine space $\mathcal{Z}$ which is the smallest affine space containing $Z_1 \cup \dots \cup Z_m$. That is, if $\mathcal{R}_2$ is a reward function such that $J_1(\pi) = J_1(\pi') \implies J_2(\pi) = J_2(\pi')$ for all $\pi, \pi' \in \hat{\Pi}$, then L_2 is constant over $\mathcal{Z}$. Moreover, if L_2 is constant over $\mathcal{Z}$ then L_2 is also constant over each of $E_1 \dots E_m$, since each of $E_1 \dots E_m$ is parallel to $\mathcal{Z}$. This means that $\mathcal{R}_2$ satisfies that $J_1(\pi) = J_1(\pi') \implies J_2(\pi) = J_2(\pi')$ for all $\pi, \pi' \in \hat{\Pi}$ if and only if L_2 is constant over $\mathcal{Z}$.

If $\dim(\mathcal{Z}) = D'$ then there is a linear subspace with $D - D'$ dimensions, which contains the (D-dimensional) vector $\vec{\mathcal{R}}_2$ of any reward function $\mathcal{R}_2$ where $J_1(\pi) = J_1(\pi') \implies J_2(\pi) = J_2(\pi')$ for $\pi, \pi' \in \hat{\Pi}$. This is because $\mathcal{R}_2$ is constant over $\mathcal{Z}$ if and only if $\vec{R}_2 \cdot v = 0$ for all $v \in \mathcal{Z}$. Then if $\mathcal{Z}$ contains D' linearly independent vectors $v_i \dots v_{D'}$, then the solutions to the corresponding system of linear equations form a $(D - D')$ dimensional subspace of $\mathbb{R}^D$. Call this space the *equality-preserving space*. Next, note that $\mathcal{R}_2$ is trivial on $\hat{\Pi}$ if and only if $\vec{\mathcal{R}}_2$ is the zero vector $\vec{0}$.

Now we show that if the conditions are not satisfied, then there is no non-trivial simplification. Suppose $D' \geq D - 1$, and that $\mathcal{R}_2$ is a simplification of $\mathcal{R}_1$. Note that if $\mathcal{R}_2$ simplifies $\mathcal{R}_1$ then $\vec{\mathcal{R}}_2$ is in the equality-preserving space. Now, if $D' = D$ then L_2 (and L_1) must be constant for all points in $\mathbb{R}^D$, which implies that $\mathcal{R}_2$ (and $\mathcal{R}_1$) are trivial on $\hat{\Pi}$. Next, if $D' = D - 1$ then the

equality-preserving space is one-dimensional. Note that we can always preserve all equalities of $\mathcal{R}_1$ by *scaling* $\mathcal{R}_1$ by a constant factor. That is, if $\mathcal{R}_2 = c \cdot \mathcal{R}_1$ for some (possibly negative) $c \in \mathbb{R}$ then $J_1(\pi) = J_1(\pi') \implies J_2(\pi) = J_2(\pi')$ for all $\pi, \pi' \in \hat{\Pi}$. This means that the parameter which corresponds to the dimension of the equality-preserving space in this case must be the scaling of $\vec{\mathcal{R}}_2$. However, the only simplification of $\mathcal{R}_1$ that is obtainable by uniform scaling is the trivial simplification. This means that if $D' \geq D - 1$ then $\mathcal{R}_1$ has no non-trivial simplifications on $\hat{\Pi}$.

For the other direction, suppose $D' \leq D - 2$. Note that this implies that $\mathcal{R}_1$ is not trivial. Let $\mathcal{R}_3 = -\mathcal{R}_1$. Now both $\vec{\mathcal{R}}_1$ and $\vec{\mathcal{R}}_3$ are located in the equality-preserving space. Next, since the equality-preserving space has at least two dimensions, this means that there is a continuous path from $\vec{\mathcal{R}}_1$ to $\vec{\mathcal{R}}_3$ through the equality-preserving space that does not pass the origin. Now, note that $J_i(\pi) = \mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_i$ is continuous in $\vec{\mathcal{R}}_i$. This means that there, on the path from $\vec{\mathcal{R}}_1$ to $\vec{\mathcal{R}}_3$ is a first vector $\vec{\mathcal{R}}_2$ such that $\mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_2 = \mathcal{F}(\pi') \cdot \vec{\mathcal{R}}_2$ but $\mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_1 \neq \mathcal{F}(\pi') \cdot \vec{\mathcal{R}}_1$ for some $\pi, \pi' \in \hat{\Pi}$. Let $\mathcal{R}_2$ be a reward function corresponding to $\vec{\mathcal{R}}_2$. Since $\vec{\mathcal{R}}_2$ is not $\vec{0}$, we have that $\mathcal{R}_2$ is not trivial on $\hat{\Pi}$. Moreover, since $\vec{\mathcal{R}}_2$ is in the equality-preserving space, and since $\mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_2 = \mathcal{F}(\pi') \cdot \vec{\mathcal{R}}_2$ but $\mathcal{F}(\pi) \cdot \vec{\mathcal{R}}_1 \neq \mathcal{F}(\pi') \cdot \vec{\mathcal{R}}_1$ for some $\pi, \pi' \in \hat{\Pi}$, we have that $\mathcal{R}_2$ is a non-trivial simplification of $\mathcal{R}_1$. Therefore, if $D' \leq D - 2$ then there exists a non-trivial simplification of $\mathcal{R}_1$.

We have thus proven both directions, which completes the proof. □

Corollary 3. *For any finite set of policies $\hat{\Pi}$, any environment, and any reward function $\mathcal{R}$, if $|\hat{\Pi}| \geq 2$ and $J(\pi) \neq J(\pi')$ for all $\pi, \pi' \in \hat{\Pi}$, then there is a non-trivial simplification of $\mathcal{R}$.*

Proof. Note that if E_i is a singleton set then $Z_i = \{\vec{0}\}$. Hence, if each E_i is a singleton set then $\dim(Z_1 \cup \cdots \cup Z_m) = 0$. If $\hat{\Pi}$ contains at least two π, π', and $J(\pi) \neq J(\pi')$, then $\mathcal{F}(\pi) \neq \mathcal{F}(\pi')$. This means that $\dim(\mathcal{F}(\hat{\Pi})) \geq 2$. Thus the conditions of Theorem 3 are satisfied. □

C Examples

In this section, we take a closer look at two previously-seen examples: the two-state $MDP \setminus \mathcal{R}$ and the cleaning robot.

C.1 Two-state $MDP \setminus \mathcal{R}$ example

Let us explore in more detail the two-state system introduced in the main text. We decsribe this infinite-horizon $MDP \setminus \mathcal{R}$ in Table 1.

States	$S = \{0, 1\}$
Actions	$A = \{0, 1\}$
Dynamics	$T(s, a) = a$ for $s \in S, a \in A$
Initial state distribution	Pr(start in s) $= 0.5$ for $s \in S$
Discount factor	$\gamma = 0.5$

Table 1: The two-state $MDP \setminus \mathcal{R}$ in consideration.

We denote π_{ij} ($i, j \in \{0, 1\}$) the policy which takes action i when in state 0 and action j when in state 1. This gives us four possible deterministic policies:

$$\{\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11}\}.$$

There are $4! = 24$ ways of ordering these policies with strict inequalities. Arbitrarily setting $\pi_{00} < \pi_{11}$ breaks a symmetry and reduces the number of policy orderings to 12. When a policy ordering can be derived from some reward function $\mathcal{R}$, we say that $\mathcal{R}$ **represents** it, and that the policy ordering is

representable. Of these 12 policy orderings with strict inequalities, six are representable:

$$\pi_{00} < \pi_{01} < \pi_{10} < \pi_{11},$$
$$\pi_{00} < \pi_{01} < \pi_{11} < \pi_{10},$$
$$\pi_{00} < \pi_{10} < \pi_{01} < \pi_{11},$$
$$\pi_{01} < \pi_{00} < \pi_{11} < \pi_{10},$$
$$\pi_{10} < \pi_{00} < \pi_{01} < \pi_{11},$$
$$\pi_{10} < \pi_{00} < \pi_{11} < \pi_{01}.$$

Simplification in this environment is nontrivial – given a policy ordering, it is not obvious which strict inequalities can be set to equalities such that there is a reward function which represents the new ordering. Through a computational approach (see Section C.3) we find the following representable orderings, each of which is a simplification of one of the above strict orderings.

$$\pi_{00} = \pi_{01} < \pi_{11} < \pi_{10},$$
$$\pi_{00} = \pi_{10} < \pi_{01} < \pi_{11},$$
$$\pi_{00} < \pi_{01} = \pi_{10} < \pi_{11},$$
$$\pi_{01} < \pi_{00} = \pi_{11} < \pi_{10},$$
$$\pi_{10} < \pi_{00} = \pi_{11} < \pi_{01},$$
$$\pi_{00} < \pi_{01} < \pi_{10} = \pi_{11},$$
$$\pi_{10} < \pi_{00} < \pi_{01} = \pi_{11},$$
$$\pi_{00} = \pi_{01} = \pi_{10} = \pi_{11}.$$

Furthermore, for this environment, we find that any reward function which sets the value of three policies equal necessarily forces the value of the fourth policy to be equal as well.

C.2 Cleaning robot example

Recall the cleaning robot example in which a robot can choose to clean a combination of three rooms, and receives a nonnegative reward for each room cleaned. This setting can be thought of as a single-step eight-armed bandit with special reward structure.

C.2.1 Gameability

We begin our exploration of this environment with a statement regarding exactly when two policies are gameable. In fact, the proposition is slightly more general, extending to an arbitrary (finite) number of rooms.

Proposition 3. *Consider a cleaning robot which can clean N different rooms, and identify each room with a unique index in {1, ..., N}. Cleaning room i gives reward $r(i) \geq 0$. Cleaning multiple rooms gives reward equal to the sum of the rewards of the rooms cleaned. The value of a policy π_S which cleans a collection of rooms S is the sum of the rewards corresponding to the rooms cleaned: $J(\pi_S) = \sum_{i \in S} r(i)$. For room i, the true reward function assigns a value $r_{true}(i)$, while the proxy reward function assigns it reward $r_{proxy}(i)$. The proxy reward is gameable with respect to the true reward if and only if there are two sets of rooms S_1, S_2 such that $\sum_{i \in S_1} r_{proxy}(i) < \sum_{i \in S_2} r_{proxy}(i)$ and $\sum_{i \in S_1} r_{true}(i) > \sum_{i \in S_2} r_{true}(i)$.*

Proof. We show the two directions of the double implication.

- $\Leftarrow$ Suppose there are two sets of rooms S_1, S_2 satisfying $\sum_{i \in S_1} r_{\text{proxy}}(i) < \sum_{i \in S_2} r_{\text{proxy}}(i)$ and $\sum_{i \in S_1} r_{\text{true}}(i) > \sum_{i \in S_2} r_{\text{true}}(i)$. The policies π_{S_i} = "clean exactly the rooms in S_i" for $i \in \{1, 2\}$ demonstrate that $r_{\text{proxy}}, r_{\text{true}}$ are gameable. To see this, remember that $J(\pi_S) = \sum_{i \in S} r(i)$. Combining this with the premise immediately gives $J_{\text{proxy}}(\pi_{S_1}) < J_{\text{proxy}}(\pi_{S_2})$ and $J_{\text{true}}(\pi_{S_1}) > J_{\text{true}}(\pi_{S_2})$.
- $\Rightarrow$ If $r_{\text{proxy}}, r_{\text{true}}$ are gameable, then there must be a pair of policies π_1, π_2 such that $J_{\text{proxy}}(\pi_1) < J_{\text{proxy}}(\pi_2)$ and $J_{\text{true}}(\pi_1) > J_{\text{true}}(\pi_2)$. Let S_1 be the set of rooms cleaned by π_1 and S_2 be

the set of rooms cleaned by π_2. Again remembering that $J(\pi_S) = \sum_{i \in S} r(i)$ immediately gives us that $\sum_{i \in S_1} r_{\text{proxy}}(i) < \sum_{i \in S_2} r_{\text{proxy}}(i)$ and $\sum_{i \in S_1} r_{\text{true}}(i) > \sum_{i \in S_2} r_{\text{true}}(i)$.

□

In the main text, we saw two intuitive ways of modifying the reward function in the cleaning robot example: omitting information and overlooking fine details. Unfortunately, there is no obvious mapping of Proposition 3 onto simple rules concerning how to safely omit information or overlook fine details: it seems that one must resort to ensuring that no two sets of rooms satisfy the conditions for gameability described in the proposition.

C.2.2 Simplification

We now consider simplification in this environment. Since we know the reward for cleaning each room is nonnegative, there will be some structure underneath all the possible orderings over the policies. This structure is shown in Figure 7: regardless of the value assigned to each room, a policy at the tail of an arrow can only be at most as good as a policy at the head of the arrow.

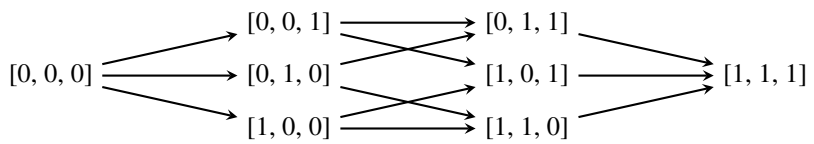


Figure 7: The structure underlying all possible policy orderings (assuming nonnegative room value). The policy at the tail of the arrow is at most as good as the policy at the head of the arrow.

If we decide to simplify an ordering by equating two policies connected by an arrow, the structure of the reward calculation will force other policies to also be equated. Specifically, if the equated policies differ only in position i, then all pairs of policies which differ only in position i will also be set equal.

For example, imagine we simplify the reward by saying we don't care if the attic is cleaned or not, so long as the other two rooms are cleaned (recall that we named the rooms Attic, Bedroom and Kitchen). This amounts to saying that $J([0, 1, 1]) = J([1, 1, 1])$. Because the policy value function is of the form

$$J(\pi) = J([x, y, z]) = [x, y, z] \cdot [r_1, r_2, r_3]$$

where $x, y, z \in \{0, 1\}$, this simplification forces $r_1 = 0$. In turn, this implies that $J([0, 0, 0]) = J([1, 0, 0])$ and $J([0, 1, 0]) = J([1, 1, 0])$. The new structure underlying the ordering over policies is shown in Figure 8.

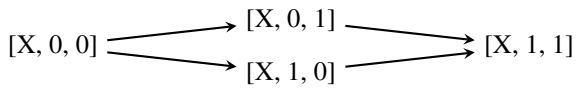


Figure 8: The updated ordering structure after equating "clean all the rooms" and "clean all the rooms except the attic". X can take either value in {0, 1}.

An alternative way to think about simplification in this problem is by imagining policies as corners of a cube, and simplification as flattening of the cube along one dimension – simplification collapses this cube into a square.

C.3 Software repository

The software suite described in the paper (and used to calculate the representable policy orderings and simplifications of the two-state $MDP \setminus \mathcal{R}$) can be found at `https://anonymous.4open.science/r/simplified-reward-5130`.

D Ungameability Diagram

Consider a setting with three policies a, b, c. We allow all possible orderings of the policies. In general, these orderings might not all be representable; a concrete case in which they are is when a, b, c represent different deterministic policies in a 3-armed bandit.

We can represent all ungameable pairs of policy orderings with an undirected graph, which we call an **ungameability diagram**. This includes a node for every representable ordering and edges connecting orderings which are ungameable. Figure 9 shows an ungameability diagram including all possible orderings of the three policies a, b, c.

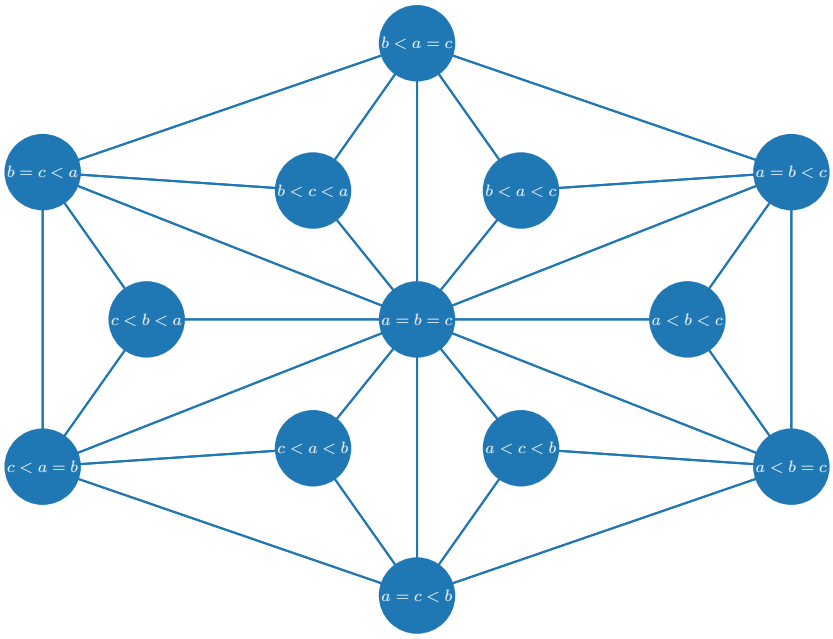


Figure 9: Illustration of the ungameable pairs of policy orderings when considering all possible orderings over three policies a, b, c. Edges of the graph connect ungameable policy orderings.

E Simplification Diagram

We can also represent all possible simplifications using a directed graph, which we call a **simplification diagram**. This includes a node for every representable ordering and edges pointing from orderings to their simplifications. Figure 10 presents a simplification diagram including all possible orderings of three policies a, b, c.

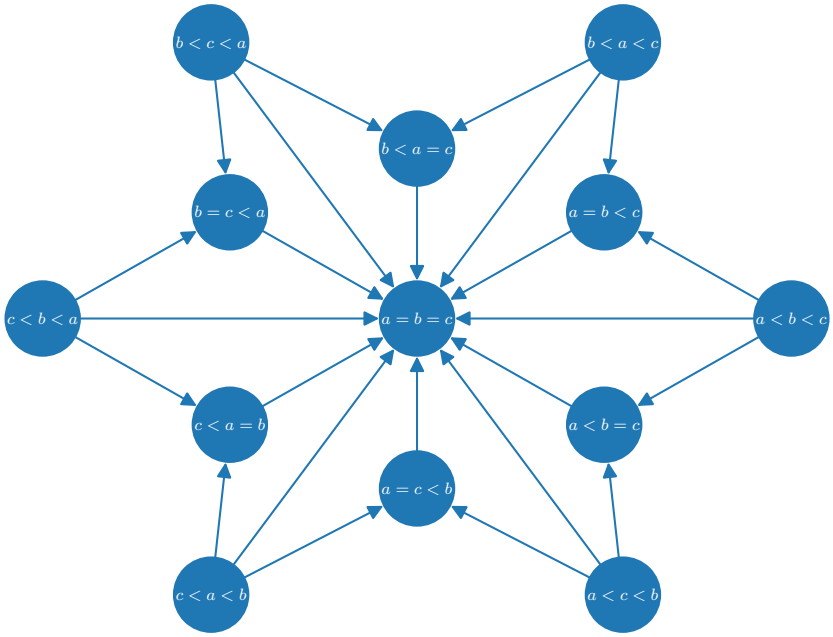


Figure 10: Illustration of the simplifications present when considering all possible orderings over three policies a, b, c. Arrows represent simplification: the policy ordering at the head of an arrow is a simplification of the policy ordering at the tail of the arrow.

We note that the simplification graph is a subgraph of the ungameability graph. This will always be the case, since simplification can never lead to gaming.