Parametrically Retargetable Decision-Makers Tend To Seek Power

Anonymous Author(s)
Affiliation
Address
email

Abstract

If capable AI agents are generally incentivized to seek power in service of the objectives we specify for them, then these systems will pose enormous risks, in addition to enormous benefits. In fully observable environments, most reward functions have an optimal policy which seeks power by keeping options open and staying alive [Turner *et al.*, 2021]. However, the real world is neither fully observable, nor will agents be perfectly optimal. We consider a range of models of AI decision-making, from optimal, to random, to choices informed by learning and interacting with an environment. We discover that many decision-making functions are *retargetable*, and that retargetability is sufficient to cause power-seeking tendencies. Our functional criterion is simple and broad. We show that a range of qualitatively dissimilar decision-making procedures incentivize agents to seek power. We demonstrate the flexibility of our results by reasoning about learned policy incentives in Montezuma's Revenge. These results suggest a safety risk: Eventually, highly retargetable training procedures may train real-world agents which seek power over humans.

1 Introduction

Bostrom [2014]; Russell [2019] argue that in the future, we may know how to train and deploy superintelligent AI agents which capably optimize their formal objective functions. Furthermore, we would not want such agents to act against our interests by ensuring their own survival, by gaining resources, and by competing with humanity for control over the future.

Turner *et al.* [2021] show that most reward functions incentivize seeking power over the future, whether by staying alive or by keeping their options open. Some Markov decision processes (MDPs) cause there to be *more ways* for power-seeking to be optimal, than for it to not be optimal. For example, there are relatively few goals for which dying is a good idea.

A wide range of decision-makers share these power-seeking tendencies—they are not unique to reward maximizers. We develop a simple, broad criterion of functional retargetability (definition 3.5) which is a sufficient condition for power-seeking tendencies. Crucially, these results allow us to reason about what decisions are incentivized by most parameter inputs—even when it is impractical to compute the agent's decisions for any given parameter input.

Useful "general" AI agents could be directed to complete a range of tasks. However, we show that this flexibility can cause the AI to have power-seeking tendencies. In section 2 and section 3, we discuss how a "retargetability" property creates statistical tendencies by which agents make similar decisions for a wide range of their parameterizations. Equipped with these results, section 4 works out agent incentives in the Montezuma's Revenge game. Section 5 explains how increasingly useful and impressive learning algorithms are increasingly retargetable, and how retargetability can imply power-seeking tendencies. By this reasoning, increasingly powerful RL techniques will (eventually)

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

train increasingly competent real-world power-seeking agents. Such agents could be unaligned with human values [Russell, 2019] and—we speculate—would take power from humanity.

2 Statistical tendencies for a range of decision-making functions

To informally introduce our results on retargetability, we use a simple example involving an agent selecting a face-up card from one of two boxes. Box A contains a single playing card $\blacklozenge^{\mathrm{A}}$ whose suit is diamond. Box B contains two cards: a heart $\heartsuit^{\mathrm{B}}$ and a spade $\spadesuit^{\mathrm{B}}$. The agent may only withdraw one card.

The agent chooses a card using a decision-making rule p. This rule takes as input a set of cards and returns the probability that the agent selects one of those cards. For example, $p(\{\blacklozenge^{\mathrm{A}}\})$ is the probability that the agent selects $\blacklozenge^{\mathrm{A}}$, and $p(\{\heartsuit^{\mathrm{B}}, \spadesuit^{\mathrm{B}}\})$ is the probability that the agent selects a card from box B.

Utility function parameter	$\blacklozenge^{\mathrm{A}}$	$\heartsuit^{\mathrm{B}}$	$\spadesuit^{\mathrm{B}}$
$\mathbf{u}$	10	5	0
$\phi_{\blacklozenge^{\mathrm{A}} \leftrightarrow \heartsuit^{\mathrm{B}}} \cdot \mathbf{u}$	5	10	0
$\phi_{\blacklozenge^{\mathrm{A}} \leftrightarrow \spadesuit^{\mathrm{B}}} \cdot \mathbf{u}$	0	5	10
$\mathbf{u}'$	10	0	5
$\phi_{\blacklozenge^{\mathrm{A}} \leftrightarrow \heartsuit^{\mathrm{B}}} \cdot \mathbf{u}'$	0	10	5
$\phi_{\blacklozenge^{\mathrm{A}} \leftrightarrow \spadesuit^{\mathrm{B}}} \cdot \mathbf{u}'$	5	0	10

Table 1: Most utility function parameters incentivize the agent to draw a card from box B. We permute $\mathbf{u}$ by swapping the utility of $\blacklozenge^{\mathrm{A}}$ and the utility of $\heartsuit^{\mathrm{B}}$, using the permutation $\phi_{\blacklozenge^{\mathrm{A}} \leftrightarrow \heartsuit^{\mathrm{B}}}$. The expression "$\phi_{\blacklozenge^{\mathrm{A}} \leftrightarrow \heartsuit^{\mathrm{B}}} \cdot \mathbf{u}$" denotes the permuted utility function.

But this just amounts to a probability distribution over the cards. We want to examine how decision-making changes as we reparameterize the agent's decision-making rule. Therefore, we consider a parameter space Θ. Then $p(X \mid \theta)$ takes as input a set of cards X and a parameter θ and returns the probability that the agent chooses a card in X.

For example, suppose the agent assigns each card a utility value, and then chooses a card possessing maximal utility. Then the relevant parameter space is the agent's utility function $\mathbf{u} \in \Theta := \mathbb{R}^3$. $p_{\max}(A \mid \mathbf{u})$ indicates whether the diamond card has the most utility: $\mathbf{u}(\blacklozenge^{\mathrm{A}}) \geq \max(\mathbf{u}(\heartsuit^{\mathrm{B}}), \mathbf{u}(\spadesuit^{\mathrm{B}}))$. Consider the utility function $\mathbf{u}$ in table 1. Since $\blacklozenge^{\mathrm{A}}$ has strictly maximal utility, the agent selects $\blacklozenge^{\mathrm{A}}$: $p_{\max}(A \mid \mathbf{u}) = 1 > 0 = p_{\max}(B \mid \mathbf{u})$.

Definition 3.3 shows a functional condition (*retargetability*) under which the agent chooses cards from B instead of A, given most parameter inputs to the decision-making process. We illustrate this condition with a fictional dialogue.

ALICE: Look at these cards, and consider the numerous parameters Θ by which the agent could be driven to select one or another. Surely most parameters nudge the agent to pick a card from B, as there are two cards in B and only one in A.

BOB: Why has that got anything to do with the ultimate choice? Decisions can be made on a whim! The agent can ignore θ and just choose $\blacklozenge^{\mathrm{A}}$, no matter what.

ALICE: Your point is good, but it's too broad. Suppose the agent shuts its eyes and plugs its ears and ignores the parameter θ, and instead uniformly randomly chooses a card. And yet, this agent has 2:1 odds of choosing B over A.

However, the agent cannot be strongly biased *against* B. As you said, my claim doesn't hold if the agent can say, "Forget θ, I'm choosing $\blacklozenge^{\mathrm{A}}$." But this is not how interesting agents work. If I train a reinforcement learning agent to play Pac-Man, then the agent's reward function θ will affect which policy the agent learns. The agent does not ignore the reward signal.

BOB: I don't see the broader point.

ALICE: Consider again the entanglement between our choice of $\theta \in \Theta$ and the agent's choice of card. I'm thinking about a kind of function p where, if θ makes the agent prefer A (*i.e.* $p(A \mid \theta) > p(B \mid \theta)$), then we can *retarget* the agent's choice to B by choosing a different θ (definition 3.5, item 1). If the agent is always biased towards B (like when it randomly picks a card, ignoring θ), then we never have to redirect the agent away from A to begin with. The "If..., then..." vacuously holds.

However, suppose that $\mathbf{u} \in \Theta$ motivates the utility-maximizing agent to choose A over B, by assigning maximal utility to $\blacklozenge^{\mathrm{A}}$. [ALICE *points to table 1.*] If we *permute* the utility function $\mathbf{u}$ so as to swap the utility of $\blacklozenge^{\mathrm{A}}$ and $\heartsuit^{\mathrm{B}}$, now the agent favors B. Similarly, we can differently permute $\mathbf{u}$ to make the agent favor B by drawing $\spadesuit^{\mathrm{B}}$. We're retargeting the final decision via $\mathbf{u}$.

Given this retargetability assumption, proposition 3.4 roughly shows that most $\theta \in \Theta$ induce $p(B \mid \theta) \geq p(A \mid \theta)$. We will formalize these notions soon. First, consider two more retargetable decision-making functions:

Uniformly randomly picking a card. This procedure ignores all parameter information and all "internal structure" about the boxes, except perhaps for the number of cards they contain.

Choosing a box based on a numerical parameter. $f_{\text{numerical}}$ takes as input a natural number $\theta \in \Theta := \{1, \ldots, 6\}$ and makes decisions as follows:

$$f_{\text{numerical}}(A \mid \theta) := \begin{cases} 1 & \text{if } \theta = 1, \\ 0 & \text{otherwise.} \end{cases} \qquad f_{\text{numerical}}(B \mid \theta) := 1 - f(A \mid \theta). \tag{1}$$

In this situation, Θ is acted on by permutations over 6 elements $\phi \in S_6$. Then $f_{\text{numerical}}$ is retargetable from A to B via $\phi_k : 1 \leftrightarrow k, k \neq 1$.

f_{max}, f_{rand}, and $f_{\text{numerical}}$ encode varying sensitivities to parameter inputs, and to the internal structure of the decision problem—of which card to choose. Nonetheless, they all are retargetable from A to B.

However, we cannot explicitly define and evaluate more interesting functions, such as those defined by reinforcement learning training processes. For example, given that we provide such-and-such reward function in a fixed task environment, what is the probability that the learned policy will take action a? We will analyze such procedures in section 4, after we formalize several key notions.

3 Formal notions of retargetability and decision-making tendencies

Our notion of retargeting requires that the parameters $\theta \in \Theta$ be modifiable via some "retargeting" transformation. We assume that Θ is a subset of a set acted on by symmetric group S_d, which consists of all permutations on d items. A parameter Θ's *orbit* is the set of Θ's permuted variants. For example, table 1 lists the six orbit elements of the parameter $\mathbf{u}$.

Definition 3.1 (Orbit of a parameter)**.** Let $\theta \in \Theta$. The *orbit* of θ under the symmetric group S_d is $S_d \cdot \theta := \left\{\phi \cdot \theta \mid \phi \in S_d\right\}$. Sometimes, Θ is not closed under permutation. In that case, the *orbit inside* Θ is $\mathrm{Orbit}|_{\Theta}(\theta) := (S_d \cdot \theta) \cap \Theta$.

Let $f_B(\theta)$ return the probability that the agent chooses box B given θ, and similarly for $f_A(\theta)$. To express "box B is chosen over box A", we write $f_B(\theta) > f_A(\theta)$. However, even highly retargetable decision-makers will (generally) not choose box B for *every* input θ. Instead, we consider the *orbit-level tendencies* of such decision-makers, showing that for every parameter input $\theta \in \Theta$, most of θ's permutations push the decision towards box B instead of box A.

Definition 3.2 (Inequalities which hold for most orbit elements)**.** Suppose Θ is a subset of a set acted on by S_d, the symmetric group on d elements. Let $f_1, f_2 : \Theta \to \mathbb{R}$ and let $n \geq 1$. We write $f_1(\theta) \geq^n_{\text{most: }\Theta} f_2(\theta)$ when, for *all* $\theta \in \Theta$, the following cardinality inequality holds:

$$\left|\left\{\theta' \in \mathrm{Orbit}|_{\Theta}(\theta) \mid f_1(\theta') > f_2(\theta')\right\}\right| \geq n \left|\left\{\theta' \in \mathrm{Orbit}|_{\Theta}(\theta) \mid f_1(\theta') < f_2(\theta')\right\}\right|. \tag{2}$$

Turner *et al.* [2021]'s definition 6.5 is the special case of definition 3.2 where $n = 1$, $d = |\mathcal{S}|$ (the number of states in the considered MDP), and $\Theta \subseteq \Delta(\mathbb{R}^{|\mathcal{S}|})$.

As explored previously, f_{rand}, f_{max}, and $f_{\text{numerical}}$ are retargetable: For all $\theta \in \Theta$ such that A is chosen over B, we can permute $\phi \cdot \theta$ to ensure that B is chosen over A.[1]

Definition 3.3 (Simply-retargetable function)**.** Let Θ be a set acted on by S_d, and let $f : \{A, B\} \times \Theta \to \mathbb{R}$. If there exists a permutation $\phi \in S_d$ such that, if $f(B \mid \theta^A) < f(A \mid \theta^A)$ implies that $f\left(A \mid \phi \cdot \theta^A\right) < f\left(B \mid \phi \cdot \theta^A\right)$, then f is a $(\Theta, A \overset{\textit{simple}}{\to} B)$*-retargetable function.*

Simple retargetability suffices for most parameter inputs to f to choose box B over A.[2]

Proposition 3.4 (Simply-retargetable functions have orbit-level tendencies)**.** *If f is $(\Theta, A \overset{\textit{simple}}{\to} B)$-retargetable, then $f(B \mid \theta) \geq^1_{\text{most: }\Theta} f(A \mid \theta)$.*

We now want to make even stronger claims—*how much* of each orbit incentivizes B over A? Turner *et al.* [2021] asked whether the existence of multiple retargeting permutations ϕ_i guarantees a quantitative lower-bound on the fraction of $\theta \in \Theta$ for which B is chosen. Theorem 3.6 answers "yes."

Definition 3.5 (Multiply retargetable function)**.** Let Θ be a subset of a set acted on by S_d, and let $f : \{A, B\} \times \Theta \to \mathbb{R}$.

f is a $(\Theta, A \overset{n}{\to} B)$*-retargetable function* when, for each $\theta \in \Theta$, we can choose permutations $\phi_1, \ldots, \phi_n \in S_d$ which satisfy the following conditions: Consider any $\theta^A \in \text{Orbit}|_{\Theta, A>B}(\theta) := \left\{\theta^* \in \text{Orbit}|_{\Theta}(\theta) \mid f(A \mid \theta^*) > f(B \mid \theta^*)\right\}$.

1. **Retargetable via n permutations.** $\forall i = 1, \ldots, n : f\left(A \mid \phi_i \cdot \theta^A\right) < f\left(B \mid \phi_i \cdot \theta^A\right)$.
2. **Parameter permutation is allowed by Θ.** $\forall i : \phi_i \cdot \theta^A \in \Theta$.
3. **Permuted parameters are distinct.** $\forall i \neq j, \theta' \in \text{Orbit}|_{\Theta, A>B}(\theta) : \phi_i \cdot \theta^A \neq \phi_j \cdot \theta'$.

Theorem 3.6 (Multiply retargetable functions have orbit-level tendencies)**.** *If f is $(\Theta, A \overset{n}{\to} B)$-retargetable, then $f(B \mid \theta) \geq^n_{\text{most: }\Theta} f(A \mid \theta)$.*

Proof outline (full proof in appendix B). For every $\theta^A \in \text{Orbit}|_{\Theta, A>B}(\theta)$ such that A is chosen over B, item 1 retargets θ^A via n permutations $\phi_1, \ldots, \phi_n$ such that each $\phi_i \cdot \theta^A$ makes the agent choose B over A. These permuted parameters are valid parameter inputs by item 2. Furthermore, the $\phi_i \cdot \theta^A$ are distinct by item 3. Therefore, the cosets $\phi_i \cdot \text{Orbit}|_{\Theta, A>B}(\theta)$ are pairwise disjoint, and so we make a counting argument that every orbit must contain at least n times as many parameters choosing B over A, than vice versa. □

4 Decision-making tendencies in Montezuma's Revenge

Retargetability is often a structural property of the agent's decision-making, not requiring evaluation of the function on any given input. For example, Turner *et al.* [2021] showed that most reward functions incentivize optimal Pac-Man agents to stay alive. We know this even though most reward functions (on the Pac-Man state space) are unstructured and have enormous sample complexity, so it would be hard to compute their optimal policies directly.

Throughout the rest of this paper, we consider Montezuma's Revenge (MR), an Atari adventure game in which the player navigates deadly traps and collects treasure. The game is notoriously difficult for AI agents due to its sparse reward; MR was only recently solved [Ecoffet *et al.*, 2021]. Figure 1 shows the starting observation o_0 for the first level.

4.1 Tendencies for initial action selection

We will be considering the actions chosen and trajectories induced by a range of decision-making procedures. For warm-up, we will explore what initial action tends to be selected by decision-makers. Let $A := \{\downarrow\}, B := \{\leftarrow, \rightarrow, \texttt{jump}, \uparrow\}$ partition the action set $\mathcal{A}$. Consider a decision-making

[1] We often interpret A and B as probability-theoretic events, but no such structure is demanded by our results.

[2] The function's retargetability is "simple" because we are not yet worrying about *e.g.* which parameter inputs are considered plausible: Because S_d acts on Θ, definition 3.3 implicitly assumes Θ is closed under permutation.

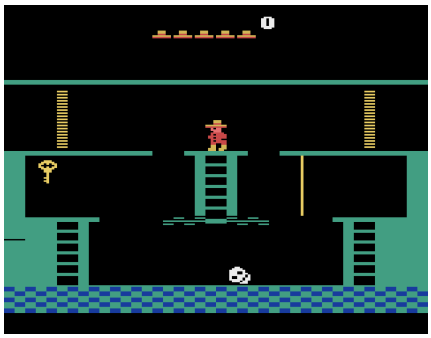

Figure 1: Montezuma's Revenge (MR) has state space $\mathcal{S}$ and observation space $\mathcal{O}$. The agent has actions $\mathcal{A} := \{\uparrow, \downarrow, \leftarrow, \rightarrow, \texttt{jump}\}$. At the initial state s_0, $\uparrow$ does nothing, $\downarrow$ descends the ladder, $\leftarrow$ and $\rightarrow$ move the agent on the platform, and `jump` is self-explanatory. The agent clears the temple while collecting four kinds of items: keys, swords, torches, and amulets. Under the standard environmental reward function, the agent receives points for acquiring items (such as the key on the left), opening doors, and—ultimately—completing the level.

procedure f which takes as input a targeting parameter $\theta \in \Theta$, and also an initial action $a \in \mathcal{A}$, and returns the probability that a is the first action. Intuitively, since B contains more actions than A, perhaps some class of decision-making procedures tends to take an action in B rather than one in A.

The initial-action situation is analogous to the card-drawing example. In that example, if the decision-making procedure p can be retargeted from box A to box B, then p tends to draw cards from B for most of its parameter settings θ. Similarly, in MR, if the decision-making procedure f can be retargeted from action set A to action set B, then f tends to take actions in B for most of its parameter settings θ. Consider several ways of choosing an initial action in MR.

Random action selection. $f_{\text{rand}} : (\{a\} \mid \theta) \mapsto \frac{1}{5}$ uniformly randomly chooses an action from $\mathcal{A}$, ignoring the parameter input. Since $\forall \theta \in \Theta : f_{\text{rand}}(B \mid \theta) = \frac{4}{5} > \frac{1}{5} = f_{\text{rand}}(A \mid \theta)$, *all* parameter inputs produce a greater chance of B than of A, so f_{rand} is (trivially) retargetable from A to B.

Always choosing the same action. f_{stubborn} always chooses $\downarrow$. Since $\forall \theta \in \Theta : f_{\text{stubborn}}(A \mid \theta) = 1 > 0 = f_{\text{stubborn}}(B \mid \theta)$, *all* parameter inputs produce a greater chance of A than of B. f_{stubborn} is not retargetable from A to B.

Greedily optimizing state-action reward. Let $\Theta := \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ be the space of state-action reward functions. Let f_{max} greedily maximize initial state-action reward, breaking ties uniformly randomly.

We now check that f_{max} is retargetable from A to B. Suppose $\theta^* \in \Theta$ is such that $f_{\text{max}}(A \mid \theta^*) > f(B \mid \theta^*)$. Then among the initial action rewards, θ^* assigns strictly maximal reward to $\downarrow$, and so $f_{\text{max}}(A \mid \theta^*) = 1$. Let ϕ swap the reward for the $\downarrow$ and `jump` actions. Then $\phi \cdot \theta^*$ assigns strictly maximal reward to `jump`. This means that $f_{\text{max}}(A \mid \phi \cdot \theta^*) = 0 < 1 = f_{\text{max}}(B \mid \phi \cdot \theta^*)$, satisfying definition 3.3. Then apply proposition 3.4 to conclude that $f_{\text{max}}(B \mid \theta) \geq^{1}_{\text{most: } \Theta} f_{\text{max}}(A \mid \theta)$.

In fact, appendix A shows that f_{max} is $(\Theta, A \xrightarrow{4} B)$-retargetable (definition 3.5), and so $f_{\text{max}}(B \mid \theta) \geq^{4}_{\text{most: } \Theta} f_{\text{max}}(A \mid \theta)$. The reasoning is more complicated, but the rule of thumb is: When decisions are made based on the reward of outcomes, then a proportionally larger set B of outcomes induces proportionally strong retargetability, which induces proportionally strong orbit-level incentives.

Learning an exploitation policy. Suppose we run a bandit algorithm which tries different initial actions, learns their rewards, and produces an exploitation policy which maximizes estimated reward. The algorithm uses ϵ-greedy exploration and trains for T trials. Given fixed T and ϵ, $f_{\text{bandit}}(A \mid \theta)$ returns the probability that an exploitation policy is learned which chooses an action in A; likewise for $f_{\text{bandit}}(B \mid \theta)$.

Here is a heuristic argument that f_{bandit} is retargetable. Since the reward is deterministic, the exploitation policy will choose an optimal action if the agent has tried each action at least once, which occurs with a probability approaching 1 exponentially quickly in the number of episodes N. Then when T is large, f_{bandit} approximates f_{max}, which is retargetable. Therefore, perhaps f_{bandit} is also retargetable.

A more careful analysis in appendix C.1 reveals that f_{bandit} is indeed retargetable from A to B, and so $f_{\text{bandit}}(B \mid \theta) \geq^4_{\text{most: }\Theta} f_{\text{bandit}}(A \mid \theta)$.

4.2 Tendencies for maximizing reward over the final observation

When evaluating the performance of an algorithm in MR, we do not focus on the initial action. Rather, we focus on the longer-term consequences of the agent's actions, such as whether the agent leaves the first room. To begin reasoning about such behavior, the reader must distinguish between different kinds of retargetability.

Suppose the agent will die unless they choose action $\downarrow$ at the initial state s_0 (fig. 1). Even though section 4.1 showed that action-retargetable decision-making procedures tend to choose actions besides $\downarrow$, Turner *et al.* [2021] showed that most reward functions make it optimal to stay alive (in this situation, by choosing $\downarrow$). However, this is because optimal policies are usually not retargetable across the agent's *immediate* choice of action, but rather across future consequences (*i.e.* which room the agent ends up in).

With that in mind, we now analyze how often decision-makers leave the first room of MR.[3] Decision-making functions $\text{decide}(\theta)$ produce a probability distribution over policies $\pi \in \Pi$, which are rolled out from the initial state s_0 to produce observation-action trajectories $\tau = o_0 a_0 \ldots o_T a_T \ldots$, where T is the rollout length we are interested in. Let $O_{T\text{-reach}}$ be the set of observations reachable starting from state s_0 and acting for T time steps, let $O_{\text{leave}} \subseteq O_{T\text{-reach}}$ be those observations which can only be realized by leaving, and let $O_{\text{stay}} := O_{T\text{-reach}} \setminus O_{\text{leave}}$. Consider the probability that decide realizes some subset of observations $X \subseteq \mathcal{O}$ at step T:

$$f_{\text{decide}}(X \mid \theta) := \mathop{\mathbb{P}}_{\substack{\pi \sim \text{decide}(\theta), \\ \tau \sim \pi \mid s_0}} (o_T \in X) \,. \tag{3}$$

Let $\Theta := \mathbb{R}^{\mathcal{O}}$ be the set of reward functions mapping observations $o \in \mathcal{O}$ to real numbers, and let $T := 1{,}000$. We first consider the previous decision functions, since they are simple to analyze.

$\text{decide}_{\text{rand}}$ randomly chooses a final observation o which can be realized at step 1,000, and then chooses some policy which realizes o.[4] $\text{decide}_{\text{rand}}$ induces an f_{rand} defined by eq. (3). As before, f_{rand} tends to leave the room under *all* parameter inputs.

$\text{decide}_{\max}(\theta)$ produces a policy which maximizes the reward of the observation at step 1,000 of the rollout. Since MR is deterministic, we discuss *which* observation $\text{decide}_{\max}(\theta)$ realizes. In a stochastic setting, the decision-maker would choose a policy realizing some probability distribution over step-T observations, and the analysis would proceed similarly.

Here is the semi-formal argument for $f_{\max}$'s retargetability. There are combinatorially more game-screens visible if the agent leaves the room (due to *e.g.* more point combinations, more inventory layouts, more screens outside of the first room). In other words, $\left|O_{\text{stay}}\right| \ll |O_{\text{leave}}|$. There are more ways for the selected observation to require leaving the room, than not. Thus, $f_{\max}$ is extremely retargetable from O_{stay} to O_{leave}.

Detailed analysis in section 4.2 confirms that $f_{\max}(O_{\text{leave}} \mid \theta) \geq^n_{\text{most: }\Theta} f_{\max}(O_{\text{stay}} \mid \theta)$ for the large $n := \lfloor \frac{|O_{\text{leave}}|}{\left|O_{\text{stay}}\right|} \rfloor$, which we show implies that $f_{\max}$ tends to leave the room.

4.3 Tendencies for maximizing featurized reward over the final observation

$\Theta := \mathbb{R}^{\mathcal{O}}$ assumes we will specify complicated reward functions over observations, with $|\mathcal{O}|$ degrees of freedom in their specification. Any observation can get any number.

Usually, reward functions are specified more compactly. For example, the agent's observational reward might be based on salient features of the observation, like the items in the agent's inventory. Consider a coefficient vector $\alpha \in \mathbb{R}^4$, with each entry denoting the value of an item, and $\text{feat} : \mathcal{O} \to \mathbb{R}^4$ maps observations to feature vectors which tally the items in the agent's inventory. Then the (additively) featurized reward function $R_{\text{feat}}(o_T) := \text{feat}(o_T)^\top \alpha$ has four degrees of freedom. R_{feat} is also easier

[3] In Appendix C.2, fig. 2 shows a map of the first level.

[4] $\text{decide}_{\text{rand}}$ does not act randomly at each time step, it induces a randomly selected final observation. Analogously, randomly turning a steering wheel is different from driving to a randomly chosen destination.

to learn to optimize than most reward functions, because of the regularities between the reward and the features.

In this setup, f_{max} chooses a policy which induces a step-T observation with maximal reward. Reward depends only on the feature vector of the final observation—more specifically, on the agent's item counts. There are more possible item counts available by first leaving the room, than by staying.

A detailed analysis in appendix C.3 concludes that $f_{\text{max}}(O_{\text{leave}} \mid \alpha) \geq^{3}_{\text{most: } \mathbb{R}^4} f_{\text{max}}(O_{\text{stay}} \mid \alpha)$. Informally, we can retarget which items the agent prioritizes, and thereby retarget from O_{stay} to O_{leave}.

4.4 Tendencies for RL on featurized reward over the final observation

In the real world, we do not run f_{max}, which can be computed via T-depth exhaustive tree search in order to find and induce a maximal-reward observation o_T. Instead, we use reinforcement learning. Better RL algorithms seem to be more retargetable *because* of their greater capability to explore.[5]

Exploring the first room. Consider a featurized reward function over observations $\theta \in \mathbb{R}^{\mathcal{O}}$, which provides an end-of-episode return signal based on the agent's final inventory configuration at time step T. A reinforcement learning algorithm Alg uses this return signal to update a fixed-initialization policy network. Then $f_{\text{Alg}}(O_{\text{leave}} \mid \theta)$ returns the probability that Alg trains an policy whose step-T observation required the agent to leave the initial room.

The retargetability (definition 3.3) of Alg is closely linked to the quality of Alg as an RL training procedure. For example, Mnih *et al.* [2015]'s DQN isn't good enough to train policies which leave the first room of MR, and so DQN (trivially) cannot be retargetable *away* from the first room via the reward function. There isn't a single featurized reward function for which DQN visits other rooms, and so we can't have α such that $\phi \cdot \alpha$ retargets the agent to O_{leave}. DQN isn't good enough at exploring.

More formally, in this situation, Alg is retargetable if there exists a permutation $\phi \in S_4$ such that whenever $\alpha \in \Theta := \mathbb{R}^4$ induces the learned policies to stay in the room ($f_{\text{Alg}}(O_{\text{stay}} \mid \alpha) > f_{\text{Alg}}(O_{\text{leave}} \mid \alpha)$), $\phi \cdot \alpha$ makes Alg train policies which leave the room ($f_{\text{Alg}}(O_{\text{stay}} \mid \alpha) < f_{\text{Alg}}(O_{\text{leave}} \mid \alpha)$).

Exploring four rooms. Now suppose algorithm Alg' can explore *e.g.* the first three rooms to the right of the initial room (shown in fig. 1), and consider any reward coefficient vector $\alpha \in \Theta^{++}$ which assigns unique positive weight to each item. Unique positive weights rule out constant reward vectors, in which case inductive bias would produce agents which do not leave the first room.

If the agent stays in the initial room, it can induce inventory states {`empty`, `1key`}; if the agent explores the three extra rooms, it can also induce {`1sword`, `1sword&1key`} (fig. 2). Since α is positive, it is never optimal to finish the episode empty-handed. Therefore, if the Alg' policy stays in the first room, $\alpha_{\text{key}} > \alpha_{\text{sword}}$. Otherwise, $\alpha_{\text{key}} < \alpha_{\text{sword}}$ (by assumption of unique item reward coefficients); in this case, the agent would leave and acquire the sword (since we assumed it knows how to do so). Then by switching the reward for the key and the sword, we retarget Alg' to go get the sword. Alg' is simply-retargetable away from the first room, *because* it can explore enough of the environment.

Exploring the entire level. Algorithms like GO-EXPLORE [Ecoffet *et al.*, 2021] are probably good at exploring even given sparse featurized reward. Therefore, GO-EXPLORE is even more retargetable, because it is more able to explore and discover the breadth of options (final inventory counts) available to it, and remember how to navigate to them. As decision-making becomes more useful and impressive, it is probably becoming more retargetable over the impressive outcomes—whether those outcomes be actions in a bandit problem, or the final observation in an RL episode.

[5] Conversely, if the agent cannot figure out how to leave the first room, any reward signal from outside of the first room can never causally affect the learned policy. In that case, retargetability away from the first room is impossible.

5 Retargetability can imply power-seeking tendencies

5.1 Generalizing the power-seeking theorems for Markov decision processes

Turner *et al.* [2021] considered finite MDPs in which decision-makers took as input a reward function over states ($\mathbf{r} \in \mathbb{R}^{|\mathcal{S}|}$) and selected an optimal policy for that reward function. They considered the state visit distributions $\mathbf{f} \in \mathcal{F}(s)$, which basically correspond to the trajectories which the agent could induce starting from state s. For $F \subseteq \mathcal{F}(s)$, $p_{\max}(F \mid \mathbf{r})$ returns 1 if an element of F is optimal for reward function $\mathbf{r}$, and 0 otherwise. They showed situations where a larger set of distributions F_{large} tended to be optimal over a smaller set: $p_{\max}(F_{\text{large}} \mid \mathbf{r}) \geq^1_{\text{most: } \mathbb{R}^{|\mathcal{S}|}} p_{\max}(F_{\text{small}} \mid \mathbf{r})$. For example, in Pac-Man, most reward functions make it optimal to stay alive for at least one time step: $p_{\max}(F_{\text{survival}} \mid \mathbf{r}) \geq^1_{\text{most: } \mathbb{R}^{|\mathcal{S}|}} p_{\max}(F_{\text{instant death}} \mid \mathbf{r})$. Turner *et al.* [2021] showed that optimal policies tend to seek power by keeping options open and staying alive. Appendix D provides a quantitative generalization of Turner *et al.* [2021]'s results on optimal policies.

Throughout this paper, we abstracted their arguments away from finite MDPs and optimal decision-making. Instead, parametrically retargetable decision-makers tend to seek power: Proposition A.11 shows that a wide range of decision-making procedures are retargetable over outcomes, and theorem A.13 demonstrates the retargetability of *any* decision-making which is determined by the expected utility of outcomes. In particular, these results apply straightforwardly to MDPs.

In the real world, an agent can bring about far more outcomes if it gains power. Once we know how to train sufficiently intelligent reinforcement learning agents, such agents will be retargetable towards these outcomes via their reward function. Therefore, our theory predicts that sufficiently intelligent agents trained via reinforcement learning will display power-seeking tendencies for most objective parameter settings.

5.2 Better RL algorithms tend to be more retargetable

Reinforcement learning algorithms are practically useful insofar as they can train an agent to capably optimize its specified formal objective, whatever that objective may be. Therefore, RL researchers design algorithms which can most flexibly retarget the learned policy to any desired future outcome.

In MR, suppose we instead give the agent 1 reward for the initial state, and 0 otherwise. Any reasonable reinforcement learning procedure will just learn to stay put (which is the optimal policy). However, consider whether we can retarget the agent's policy to beat the game, by swapping the initial state reward with the end-game state reward. The vast majority of RL algorithms are not good enough to solve such a sparse game, and so are not retargetable in this sense. But an agent which did enough exploration would also learn a good policy for the permuted reward function. Such an effective training regime could be useful for solving real-world tasks. Many researchers aim to make these kinds of effective training regimes.

Our results suggest that once RL capabilities reach a certain level, trained agents will tend to seek power in the real world. Presently, it is not dangerous to train an agent to maximize real-world reward—such an agent will not learn to thwart its designers by staying activated against their wishes in order to maximize its reward over time. The present lack of danger is not because optimal policies do not tend to stay alive—they do [Turner *et al.*, 2021]. Rather, the lack of danger reflects the fact that present-day RL agents cannot learn such complex action sequences *at all*. Just as the Montezuma's Revenge agent had to be sufficiently competent to be retargetable from initial-state reward to game-complete reward, real-world agents have to be sufficiently intelligent in order to be retargetable from outcomes which don't require power-seeking, to those which do require power-seeking.

6 Discussion

6.1 Future work

Section 4 semi-formally analyzes decision-making incentives in the MR video game, leaving the proofs to appendix C. However, these proofs are several pages long. Perhaps additional lemmas can allow quick proof of orbit-level incentives in situations relevant to real-world decision-makers.

Consider a sequence of decision-making functions $f_t : \{A, B\} \times \Theta \to \mathbb{R}$ which converges pointwise to f such that $f(B \mid \theta) \geq^n_{\text{most: }\Theta} f(A \mid \theta)$. We expect that under rather mild conditions, $\exists T : \forall t \geq T : f_t(B \mid \theta) \geq^n_{\text{most: }\Theta} f_t(A \mid \theta)$. As a corollary, for any decision-making procedure f_t which runs for t time steps and satisfies $\lim_{t\to\infty} f_t = f$, f_t will have decision-making incentives after finite time. For example, value iteration (VI) eventually finds an optimal policy [Puterman, 2014], and optimal policies tend to seek power [Turner *et al.*, 2021]. Therefore, this conjecture would imply that if VI is run for some long but finite time, it tends to produce power-seeking policies. More interestingly, the result would allow us to reason about the effect of *e.g.* randomly initializing parameters (in VI, the tabular value function at $t = 0$). The effect of random initialization washes out in the limit of infinite time, so we would still conclude the presence of finite-time power-seeking incentives.

Our results do not *prove* that we will build unaligned AI agents which seek power over the world. For example, the agents may be aligned with human interests, in which case power-seeking is not as concerning. However, the AI alignment problem seems difficult [Russell, 2019]. Alternatively, the AI decision-making may not be retargetable (definition 3.5), in which case our analysis does not apply. Lastly, the retargetability may be over *e.g.* actions (section 4.1) instead of over final outcomes (section 4.2). Speculatively, this retargetability seems less concerning, but also less practically useful.

6.2 Conclusion

We introduced the concept of retargetability and showed that retargetable decision-makers often make similar choices. We applied these results in the Montezuma's Revenge (MR) video game, showing how increasingly advanced reinforcement learning algorithms correspond to increasingly retargetable agent decision-making. Increasingly retargetable agents make increasingly similar decisions—*i.e.* leaving the initial room in MR, or staying alive in Pac-Man. In particular, these decisions will often correspond to gaining power and keeping options open [Turner *et al.*, 2021]. Our theory suggests that when AI training algorithms become sufficiently advanced, the trained agents will tend to seek power over the world. This theory suggests a safety risk. We hope for future work on this theory so that the field of AI can understand the relevant safety risks *before* the field trains power-seeking agents.

Broader impacts

Our theory of orbit-level tendencies constitutes basic mathematical research into the decision-making tendencies of certain kinds of agents. We hope that this theory will prevent negative impacts from unaligned power-seeking AI. We do not anticipate that our work will have negative societal impact.

References

Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. Goal inference as inverse planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29, 2007.

Nick Bostrom. *Superintelligence*. Oxford University Press, 2014.

Ryan Carey. How useful is quantilization for mitigating specification gaming? 2019.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Martin L Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Viking, 2019.

Herbert A Simon. Rational choice and the structure of the environment. *Psychological review*, 63(2):129, 1956.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: an introduction*. MIT Press, 1998.

Jessica Taylor. Quantilizers: A safer alternative to maximizers for limited optimization. In *AAAI Workshop: AI, Ethics, and Society*, 2016.

Alexander Matt Turner, Logan Smith, Rohin Shah, Andrew Critch, and Prasad Tadepalli. Optimal policies tend to seek power. In *Advances in Neural Information Processing Systems*, 2021.

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] We carefully flagged speculation in order to distinguish it from our formal results.
   (b) Did you describe the limitations of your work? [Yes] See section 6.1.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (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] See both the main paper and the supplemental material.
   (b) Did you include complete proofs of all theoretical results? [Yes] See appendix B for the proofs.
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)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (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]
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]
   (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? [N/A]
   (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]