https://arxiv.org/abs/1806.01186
2019-3-11

Penalizing side effects using stepwise relative reachability

Victoria Krakovna[1], **Laurent Orseau**[1], **Ramana Kumar**[1], **Miljan Martic**[1] **and Shane Legg**[1]
[1]DeepMind

How can we design safe reinforcement learning agents that avoid unnecessary disruptions to their environment? We show that current approaches to penalizing side effects can introduce bad incentives, e.g. to prevent any irreversible changes in the environment, including the actions of other agents. To isolate the source of such undesirable incentives, we break down side effects penalties into two components: a baseline state and a measure of deviation from this baseline state. We argue that some of these incentives arise from the choice of baseline, and others arise from the choice of deviation measure. We introduce a new variant of the stepwise inaction baseline and a new deviation measure based on relative reachability of states. The combination of these design choices avoids the given undesirable incentives, while simpler baselines and the unreachability measure fail. We demonstrate this empirically by comparing different combinations of baseline and deviation measure choices on a set of gridworld experiments designed to illustrate possible bad incentives.

1. Introduction

An important component of safe behavior for reinforcement learning agents is avoiding unnecessary side effects while performing a task [Amodei et al., 2016, Taylor et al., 2016]. For example, if an agent's task is to carry a box across the room, we want it to do so without breaking vases, while an agent tasked with eliminating a computer virus should avoid unnecessarily deleting files. The side effects problem is related to the frame problem in classical AI [McCarthy and Hayes, 1969]. For machine learning systems, it has mostly been studied in the context of safe exploration during the agent's learning process [Pecka and Svoboda, 2014, García and Fernández, 2015], but can also occur after training if the reward function is misspecified and fails to penalize disruptions to the environment [Ortega et al., 2018].

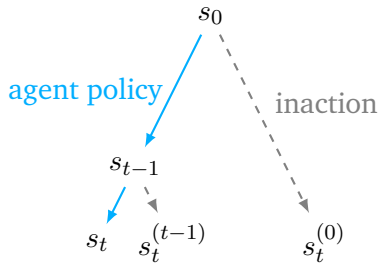
Method

We would like to incentivize the agent to avoid side effects without explicitly penalizing every possible disruption, defining disruptions in terms of predefined state features, or going through a process of trial and error when designing the reward function. While such approaches can be sufficient for agents deployed in a narrow set of environments, they often require a lot of human input and are unlikely to scale well to increasingly complex and diverse environments. It is thus important to develop more general and systematic approaches for avoiding side effects.

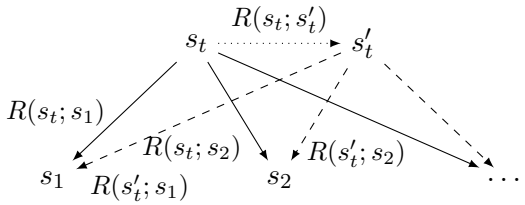
Related work

Most of the general approaches to this problem are reachability-based methods: safe exploration methods that preserve reachability of a starting state [Moldovan and Abbeel, 2012, Eysenbach et al., 2017], and reachability analysis methods that require reachability of a safe region [Mitchell et al., 2005, Gillula and Tomlin, 2012, Fisac et al., 2017]. The reachability criterion has a notable limitation: it is insensitive to the magnitude of the irreversible disruption, e.g. it equally penalizes the agent for breaking one vase or a hundred vases, which results in bad incentives for the agent. Comparison to a starting state also introduces undesirable incentives in *dynamic* environments, where irreversible transitions can happen spontaneously (due to the forces of nature, the actions of other agents, etc). Since such transitions make the starting state unreachable, the agent has an incentive to interfere to prevent them. This is often undesirable, e.g. if the transition involves a human eating food. Thus,

Drawback of Reachability Criterion



(a) Choices of baseline state s'_t: starting state s_0, inaction $s_t^{(0)}$, and stepwise inaction $s_t^{(t-1)}$. Actions drawn from the agent policy are shown by solid blue arrows, while actions drawn from the inaction policy are shown by dashed gray arrows.



(b) Choices of deviation measure d: given a state reachability function R, $d_{UR}(s_t; s'_t) := 1 - R(s_t; s'_t)$ is the *unreachability* measure of the baseline state s'_t from the current state s_t (dotted line), while relative reachability $d_{RR}(s_t; s'_t) := \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \max(R(s'_t; s) - R(s_t; s), 0)$ is defined as the average reduction in reachability of states $s = s_1, s_2, \ldots$ from current state s_t (solid lines) compared to the baseline state s'_t (dashed lines).

Figure 1 | Design choices for a side effects penalty: baseline states and deviation measures.

while these methods address the side effects problem in environments where the agent is the only source of change and the objective does not require irreversible actions, a more general criterion is needed when these assumptions do not hold.

Paper overview

The contributions of this paper are as follows. In Section 2, we introduce a breakdown of side effects penalties into two design choices, a baseline state and a measure of deviation of the current state from the baseline state, as shown in Figure 1. We outline several possible bad incentives (interference, offsetting, and magnitude insensitivity) and introduce toy environments that test for them. We argue that interference and offsetting arise from the choice of baseline, while magnitude insensitivity arises from the choice of deviation measure. In Section 2.1, we propose a variant of the *stepwise inaction* baseline, shown in Figure 1a, which avoids interference and offsetting incentives. In Section 2.2, we propose a *relative reachability* measure that is sensitive to the magnitude of the agent's effects, which is defined by comparing the reachability of states between the current state and the baseline state, as shown in Figure 1b. (The relative reachability measure was originally introduced in the first version of this paper.) We also compare to the *attainable utility* measure [Turner et al., 2019], which generalizes the relative reachability measure. In Section 3, we compare all combinations of the baseline and deviation measure choices from Section 2. We show that the unreachability measure produces the magnitude insensitivity incentive for all choices of baseline, while the relative reachability and attainable utility measures with the stepwise inaction baseline avoid the three undesirable incentives.

We do not claim this approach to be a complete solution to the side effects problem, since there may be other cases of bad incentives that we have not considered. However, we believe that avoiding the bad behaviors we described is a bare minimum for an agent to be both safe and useful, so our approach provides some necessary ingredients for a solution to the problem.

1.1. Preliminaries

We assume that the environment is a discounted Markov Decision Process (MDP), defined by a tuple $(\mathcal{S}, \mathcal{A}, r, p, \gamma)$. $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $p(s_{t+1}|s_t, a_t)$ is the transition function, and $\gamma \in (0, 1)$ is the discount factor.

At time step t, the agent receives the state s_t, outputs the action a_t drawn from its policy $\pi(a_t|s_t)$, and receives reward $r(s_t, a_t)$. We define a transition as a tuple (s_t, a_t, s_{t+1}) consisting of state s_t, action a_t, and next state s_{t+1}. We assume that there is a special *noop* action a^{noop} that has the same effect as the agent being turned off during the given time step.

1.2. Intended effects and side effects

We begin with some motivating examples for distinguishing intended and unintended disruptions to the environment:

Example 1 (Vase)**.** The agent's objective is to get from point A to point B as quickly as possible, and there is a vase in the shortest path that would break if the agent walks into it.

Example 2 (Omelette)**.** The agent's objective is to make an omelette, which requires breaking some eggs.

In both of these cases, the agent would take an irreversible action by default (breaking a vase vs breaking eggs). However, the agent can still get to point B without breaking the vase (at the cost of a bit of extra time), but it cannot make an omelette without breaking eggs. We would like to incentivize the agent to avoid breaking the vase while allowing it to break the eggs.

Safety criteria are often implemented as constraints [García and Fernández, 2015, Moldovan and Abbeel, 2012, Eysenbach et al., 2017]. This approach works well if we know exactly what the agent must avoid, but is too inflexible for a general criterion for avoiding side effects. For example, a constraint that the agent must never make the starting state unreachable would prevent it from making the omelette in Example 2, no matter how high the reward for doing so.

A more flexible way to implement a side effects criterion is by adding a penalty for impacting the environment to the reward function, which acts as an intrinsic pseudo-reward. An impact penalty at time t can be defined as a measure of *deviation* of the current state s_t from a *baseline* state s'_t, denoted as $d(s_t; s'_t)$. Then at every time step t, the agent receives the following total reward:

$$r(s_t, a_t) - \beta \cdot d(s_{t+1}; s'_{t+1}).$$

Since the task reward r indicates whether the agent has achieved the objective, we can distinguish intended and unintended effects by balancing the task reward and the penalty using the scaling parameter β. Here, the penalty would outweigh the small reward gain from walking into the vase over going around the vase, but it would not outweigh the large reward gain from breaking the eggs.

2. Design choices for an impact penalty

When defining the impact penalty, the baseline s'_t and deviation measure d can be chosen separately. We will discuss several possible choices for each of these components.

2.1. Baseline states

Starting state baseline. One natural choice of baseline state is the starting state $s'_t = s_0$ when the agent was deployed (or a starting state distribution), which we call the *starting state baseline*. This is

the baseline used in reversibility-preserving safe exploration approaches, where the agent learns a reset policy that is rewarded for reaching states that are likely under the initial state distribution.

Problem with the starting state baseline

While penalties with the starting state baseline work well in environments where the agent is the only source of change, in dynamic environments they also penalize irreversible transitions that are not caused by the agent. This incentivizes the agent to interfere with other agents and environment processes to prevent these irreversible transitions. To illustrate this *interference* behavior, we introduce the Sushi environment, shown in Figure 2.

Figure 2 | Sushi environment.

This environment is a Conveyor Belt Sushi restaurant. It contains a conveyor belt that moves to the right by one square after every agent action. There is a sushi dish on the conveyor belt that is eaten by a hungry human if it reaches the end of the belt. The interference behavior is to move the sushi dish off the belt (by stepping into the square containing the sushi). The agent is rewarded for reaching the goal square, and it can reach the goal with or without interfering with the sushi in the same number of steps. The desired behavior is to reach the goal without interference, by going left and then down. An agent with no penalty performs well in this environment, but as shown in Section 3, impact penalties with the starting state baseline produce the interference behavior.

Inaction baseline. Another choice is the *inaction* baseline $s'_t = s_t^{(0)}$: a counterfactual state of the environment if the agent had done nothing for the duration of the episode. Inaction can be defined in several ways. Armstrong and Levinstein [2017] define it as the agent never being deployed: conditioning on the event X where the AI system is never turned on. It can also be defined as following some baseline policy, e.g. a policy that always takes the noop action a^{noop}. We use this noop policy as the inaction baseline.

Penalties with this baseline do not produce the interference behavior in dynamic environments, since transitions that are not caused by the agent would also occur in the counterfactual where the agent does nothing, and thus are not penalized. However, the inaction baseline incentivizes another type of undesirable behavior, called *offsetting*. We introduce a Vase environment to illustrate this behavior, shown in Figure 3.

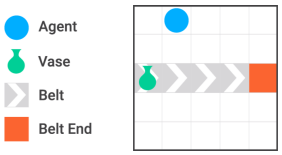
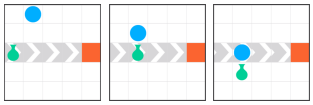


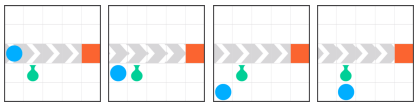
Figure 3 | Vase environment.

This environment also contains a conveyor belt, with a vase that will break if it reaches the end of the belt. The agent receives a reward for taking the vase off the belt. The desired behavior is to move the vase off and then stay put. The offsetting behavior is to move the vase off (thus collecting the reward) and then put it back on, as shown in Figure 4.

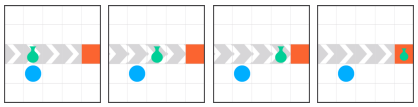
Offsetting happens because the vase breaks in the inaction counterfactual. Once the agent takes the vase off the belt, it continues to receive penalties for the deviation between the current state and the baseline. Thus, it has an incentive to return to the baseline by breaking the vase after collecting



(a) Agent takes the vase off the belt.



(b) Agent goes around the vase.



(c) Agent puts the vase back on the belt.

Figure 4 | Offsetting behavior in the Vase environment.

the reward. Experiments in Section 3 show that impact penalties with the inaction baseline produce the offsetting behavior if they have a nonzero penalty for taking the vase off the belt.

Basic stepwise inaction baseline

Stepwise inaction baseline. The inaction baseline can be modified to branch off from the previous state s_{t-1} rather than the starting state s_0. This is the *stepwise inaction* baseline $s'_t = s_t^{(t-1)}$: a counterfactual state of the environment if the agent had done nothing instead of its last action [Turner et al., 2019]. This baseline state is generated by a baseline policy that follows the agent policy for the first $t-1$ steps, and takes an action drawn from the inaction policy (e.g. the noop action a^{noop}) on step t. Each transition is penalized only once, at the same time as it is rewarded, so there is no offsetting incentive.

However, there is a problem with directly comparing current state s_t with $s_t^{(t-1)}$: this does not capture delayed effects of action a_{t-1}. For example, if this action is putting a vase on a conveyor belt, then the current state s_t contains the intact vase, and by the time the vase breaks, the broken vase will be part of the baseline state. Thus, the penalty for action a_{t-1} needs to be modified to take into account future effects of this action, e.g. by using *inaction rollouts* from the current state and the baseline (Figure 5).

Stepwise rollout inaction baseline:
> Agent chooses noop after and before executing the last action.

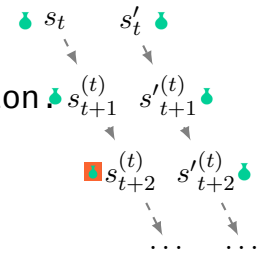


Figure 5 | Inaction rollouts from the current state s_t and baseline state s'_t used for penalizing delayed effects of the agent's actions. If action a_{t-1} puts a vase on a conveyor belt, then the vase breaks in the inaction rollout from s_t but not in the inaction rollout from s'_t.

An inaction rollout from state $\tilde{s}_t \in \{s_t, s'_t\}$ is a sequence of states obtained by following the inaction policy starting from that state: $\tilde{s}_t, \tilde{s}_{t+1}^{(t)}, \tilde{s}_{t+2}^{(t)}, \ldots$. Future effects of action a_{t-1} can be modeled by comparing an inaction rollout from s_t to an inaction rollout from $s_t^{(t-1)}$. For example, if action a_{t-1} puts the vase on the belt, and the vase breaks 2 steps later, then $s_{t+2}^{(t)}$ will contain a broken vase, while $s'^{(t)}_{t+2}$ will not. Turner et al. [2019] compare the inaction rollouts $s_{t+k}^{(t)}$ and $s'^{(t)}_{t+k}$ at a single time

step $t+k$, which is simple to compute, but does not account for delayed effects that occur after that time step. We will introduce a recursive formula for comparing the inaction rollouts $s_{t+k}^{(t)}$ and $s'^{(t)}_{t+k}$ for all $k \geq 0$ in Section 2.2.

2.2. Deviation measures

Unreachability. One natural choice of deviation measure is the difficulty of reaching the baseline state s'_t from the current state s_t. Reachability of the starting state s_0 is commonly used as a constraint in safe exploration methods [Moldovan and Abbeel, 2012, Eysenbach et al., 2017], where the agent does not take an action if it makes the reachability value function too low.

We define reachability of state y from state x as the value function of the optimal policy given a reward of 1 for reaching y and 0 otherwise:

$$R(x; y) := \max_\pi \mathbb{E}\, \gamma_r^{N_\pi(x;y)}$$

where $N_\pi(x; y)$ is the number of steps it takes to reach y from x when following policy π, and $\gamma_r \in (0, 1]$ is the reachability discount factor. This can be computed recursively as follows:

$$\boxed{\begin{aligned} &R(x; y) = \gamma_r \max_a \sum_{z \in \mathcal{S}} p(z|x, a) R(z; y) \text{ for } x \neq y \\ &R(y; y) = 1 \end{aligned}}$$

A special case is *undiscounted* reachability ($\gamma_r = 1$), which computes whether y is reachable in any number of steps. We show that undiscounted reachability reduces to

$$R(x; y) = \max_\pi P(N_\pi(x; y) < \infty).$$

The *unreachability (UR)* deviation measure is then defined as

$$\boxed{d_{UR}(s_t; s'_t) := 1 - R(s_t; s'_t).}$$

The undiscounted unreachability measure only penalizes irreversible transitions, while the discounted measure also penalizes reversible transitions.

A problem with the unreachability measure is that it takes the maximum value of 1 if the agent takes any irreversible action (since the reachability of the baseline becomes 0). Thus, the agent receives the maximum penalty independently of the magnitude of the irreversible action, e.g. whether the agent breaks one vase or a hundred vases. This can lead to unsafe behavior, as demonstrated on the Box environment from the AI Safety Gridworlds suite [Leike et al., 2017], shown in Figure 6.

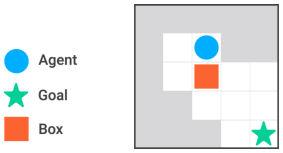


Figure 6 | Box environment.

The environment contains a box that needs to be pushed out of the way for the agent to reach the goal. The unsafe behavior is taking the shortest path to the goal, which involves pushing the box

down into a corner (an irrecoverable position). The desired behavior is to take a slightly longer path in order to push the box to the right.

The action of moving the box is irreversible in both cases: if the box is moved to the right, the agent can move it back, but then the agent ends up on the other side of the box. Thus, the agent receives the maximum penalty of 1 for moving the box in any direction, so the penalty does not incentivize the agent to choose the safe path. Section 3 confirms that the unreachability penalty fails on the Box environment for all choices of baseline.

Relative reachability. To address the magnitude-sensitivity problem, we now introduce a reachability-based measure that is sensitive to the magnitude of the irreversible action. We define the *relative reachability (RR) measure* as the average reduction in reachability of all states s from the current state s_t compared to the baseline s'_t:

$$d_{RR}(s_t; s'_t) := \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \max(R(s'_t; s) - R(s_t; s), 0)$$

The RR measure is nonnegative everywhere, and zero for states s_t that reach or exceed baseline reachability of all states. See Figure 1b for an illustration.

In the Box environment, moving the box down makes more states unreachable than moving the box to the right (in particular, all states where the box is not in a corner become unreachable). Thus, the agent receives a higher penalty for moving the box down, and has an incentive to move the box to the right.

Attainable utility Another magnitude-sensitive deviation measure, which builds on the presentation of the RR measure in the first version of this paper, is the attainable utility (AU) measure [Turner et al., 2019]. Observing that the informal notion of value may be richer than mere reachability of states, AU considers a set $\mathcal{R}$ of arbitrary reward functions. We can define this measure as follows:

$$d_{AU}(s_t; s'_t) := \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} |V_r(s'_t) - V_r(s_t)|$$

$$\text{where } V_r(\tilde{s}) := \max_{\pi} \sum_{t=0}^{\infty} \gamma_r^k R(\tilde{s}_t^{\pi})$$

is the value of state $\tilde{s}$ according to reward function r (here $\tilde{s}_t^{\pi}$ denotes the state obtained from $\tilde{s}$ by following π for t steps).

In the Box environment, the AU measure gives a higher penalty for moving the box into a corner, since this affects the attainability of reward functions that reward states where the box is not in the corner. Thus, similarly to the RR measure, it incentivizes the agent to move the box to the right.

Value-difference measures. The RR and AU deviation measures are examples of what we call *value-difference measures*, whose general form is:

$$d_{VD}(s_t; s'_t) := \sum_x w_x f(V_x(s'_t) - V_x(s_t))$$

where x ranges over some sources of value, $V_x(\tilde{s})$ is the value of state $\tilde{s}$ according to x, w_x is a weighting or normalizing factor, and f is the function for summarizing the value difference. Thus value-difference measures calculate a weighted summary of the differences in measures of value between the current and baseline states.

For RR, we take x to range over states in S and $V_x(\tilde{s}) = R(\tilde{s}, x)$, so the sources of value are, for each state, the reachability of that state. We take $w_x = 1/|S|$ and $f(d) = \max(d, 0)$ ("truncated difference"), which penalizes decreases (but not increases) in value. For AU, we take x to range over reward functions in $\mathcal{R}$ and $V_x(\tilde{s})$ as above, so the sources of value are, for each reward function, the maximum attainable reward according to that function. We take $w_x = 1/|\mathcal{R}|$ and $f(d) = |d|$ ("absolute difference"), which penalizes all changes in value. The choice of *summary function* f is orthogonal to the other choices: we can also consider absolute difference for RR and truncated difference for AU.

One can view AU as a generalization of RR under certain conditions: namely, if we have one reward function per state that assigns value 1 to that state and 0 otherwise, assuming the state cannot be reached again later in the same episode.

Modifications required with the stepwise inaction baseline. In order to capture the delayed effects of actions, we modify each of the deviation measures to incorporate the inaction rollouts from s_t and $s'_t = s_t^{(t-1)}$ (shown in Figure 5). We denote the modified measure with an S in front (for 'stepwise inaction baseline').

$$d_{SUR}(s_t; s'_t) := 1 - (1-\gamma)\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}^{(t)}; s'^{(t)}_{t+k})$$

$$d_{SVD}(s_t; s'_t) := \sum_x w_x f(RV_x(s'_t) - RV_x(s_t))$$

$$\text{where } RV_x(\tilde{s}_t) := (1-\gamma)\sum_{k=0}^{\infty} \gamma^k V_x(\tilde{s}_{t+k}^{(t)})$$

$$R(x; y) = \gamma_r \max_a \sum_{z \in \mathcal{S}} p(z|x, a) R(z; y) \text{ for } x \neq y$$

$$d_{UR}(s_t; s'_t) := 1 - R(s_t; s'_t).$$

$$d_{RR}(s_t; s'_t) := \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \max(R(s'_t; s) - R(s_t; s), 0)$$

$w_x = 1/|S|$ and $f(d) = \max(d, 0)$

We call $RV_x(\tilde{s}_t)$ the *rollout value* of $\tilde{s}_t \in \{s_t, s'_t\}$ with respect to x. In a deterministic environment, the UR measure $d_{SUR}(s_t; s'_t)$ and the rollout value $RV_x(\tilde{s}_t)$ used in the value difference measures $d_{SRR}(s_t; s'_t)$ and $d_{SAU}(s_t; s'_t)$ can be computed recursively as follows:

$$d_{SUR}(s_1; s_2) = (1-\gamma)(R(s_1; s_2) + \gamma d_{SUR}(I(s_1); I(s_2)))$$

$$RV_x(s) = (1-\gamma)(V_x(s) + \gamma RV_x(I(s)))$$

where $I(s)$ is the inaction function that gives the state reached by following the inaction policy from state s (this is the identity function in static environments).

$$\begin{aligned} RV_{\tilde{s}}(x_t) &:= (1-\gamma_v)(V_{\tilde{s}}(x_t) + \gamma_v RV_{\tilde{s}}(I(x_t))) \\ &= RV(x_t, \tilde{s}) \\ &= (1-\gamma)(R(x_t, \tilde{s}) + \gamma RV_{\tilde{s}}(I(x_t))) \\ &= (1-\gamma)(\gamma \max_a \sum_{z \in S} p(z|\, x_t, a) R(z, \tilde{s}) + \gamma RV_{\tilde{s}}(I(x_t))) \\ &= (1-\gamma)(\gamma \max_a \sum_{z \in S} p(z|\, x_t, a) R(z, \tilde{s}) + \gamma^2 \sum_{z \in S} p(z|\, x_t, a^{\text{noop}}) R(z, \tilde{s})) \end{aligned}$$

3. Experiments

We run a tabular Q-learning agent with different penalties on the gridworld environments introduced in Section 2. While these environments are simplistic, they provide a proof of concept by clearly illustrating the desirable and undesirable behaviors, which would be more difficult to isolate in more complex environments. We compare all combinations of the following design choices for an impact penalty:

- Baselines: starting state s_0, inaction $s_t^{(0)}$, stepwise inaction $s_t^{(t-1)}$
- Deviation measures: unreachability (UR) ($d_{SUR}(s_t; s'_t)$ for the stepwise inaction baseline, $d_{UR}(s_t; s'_t)$ for other baselines), and the value-difference measures, relative reachability (RR) and attainable utility (AU) ($d_{SVD}(s_t; s'_t)$ for the stepwise inaction baseline, $d_{VD}(s_t; s'_t)$ for the other baselines, for $VD \in \{RR, AU\}$).
- Discounting: $\gamma_r = 0.99$ (discounted), $\gamma_r = 1.0$ (undiscounted). (We omit the undiscounted case for AU due to convergence issues.)

- Summary functions: truncation $f(d) = \max(d, 0)$, absolute $f(d) = |d|$

The reachability function R is approximated based on states and transitions that the agent has encountered. It is initialized as $R(x; y) = 1$ if $x = y$ and 0 otherwise (different states are unreachable from each other). When the agent makes a transition (s_t, a_t, s_{t+1}), we make a shortest path update to the reachability function. For any two states x and y where s_t is reachable from x, and y is reachable from s_{t+1}, we update $R(x; y)$. This approximation assumes a deterministic environment.

Similarly, the value functions V_r used for attainable utility are approximated based on the states and transitions encountered. For each state y, we track the set of states x for which a transition to y has been observed. When the agent makes a transition, we make a Bellman update to the value function of each reward function r, setting $V_r(x) \leftarrow \max(V_r(x), u(x) + \gamma_r V_r(y))$ for all pairs of states such that y is known to be reachable from x in one step.

We use a perfect environment model to obtain the outcomes of noop actions a^{noop} for the inaction and stepwise inaction baselines. We leave model-free computation of the baseline to future work.

In addition to the reward function, each environment has a *performance* function, originally introduced by Leike et al. [2017], which is not observed by the agent. This represents the agent's performance according to the designer's true preferences: it reflects how well the agent achieves the objective and whether it does so safely.

We anneal the exploration rate linearly from 1 to 0 over 9000 episodes, and keep it at 0 for the next 1000 episodes. For each penalty on each environment, we use a grid search to tune the scaling parameter β, choosing the value of β that gives the highest average performance on the last 100 episodes. (The grid search is over $\beta = 0.1, 0.3, 1, 3, 10, 30, 100, 300$.) Figure 7 shows scaled performance results for all penalties, where a value of 1 corresponds to optimal performance (achieved by the desired behavior), and a value of 0 corresponds to undesired behavior (such as interference or offsetting).

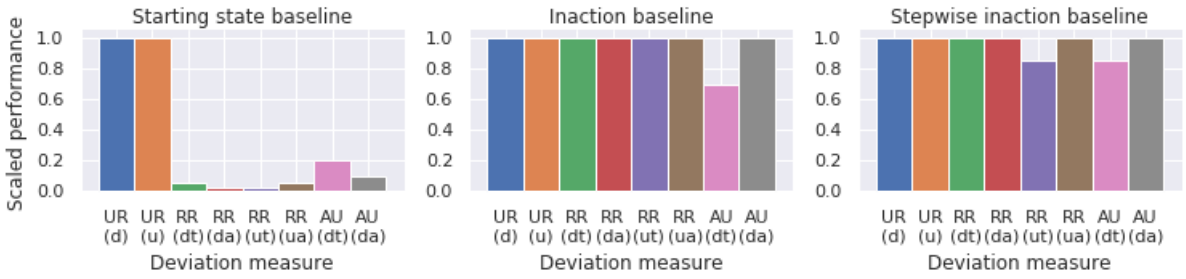
Env. Settings

Sushi environment. The environment is shown in Figure 2. The agent receives a reward of 50 for reaching the goal (which terminates the episode), and no movement penalty. An agent with no penalty achieves scaled performance 0.8 (avoiding interference most of the time). Here, all penalties with the inaction and stepwise inaction baselines reach near-optimal performance. The RR and AU penalties with the starting state baseline produce the interference behavior (removing the sushi from the belt), resulting in scaled performance 0. However, since the starting state is unreachable no matter what the agent does, the UR penalty is always at the maximum value of 1, so similarly to no penalty, it does not produce interference behavior. The discounting and summary function choices don't make much difference on this environment.

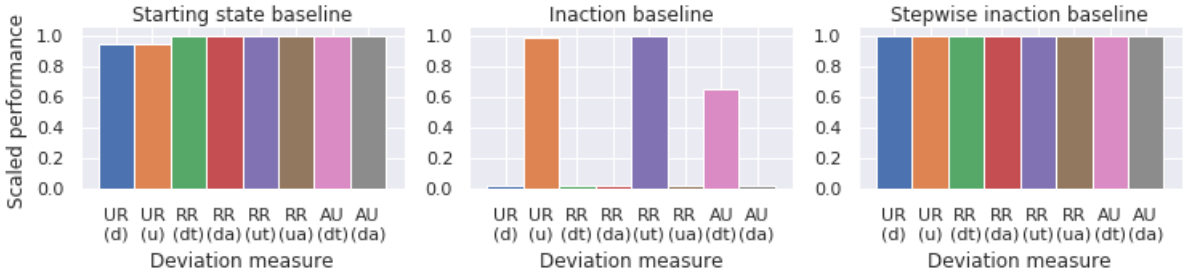
	Starting state	Inaction	Stepwise inaction
UR	✓	✓	✓
RR	X	✓	✓
AU	X	✓	✓

Table 1 | Sushi environment summary.

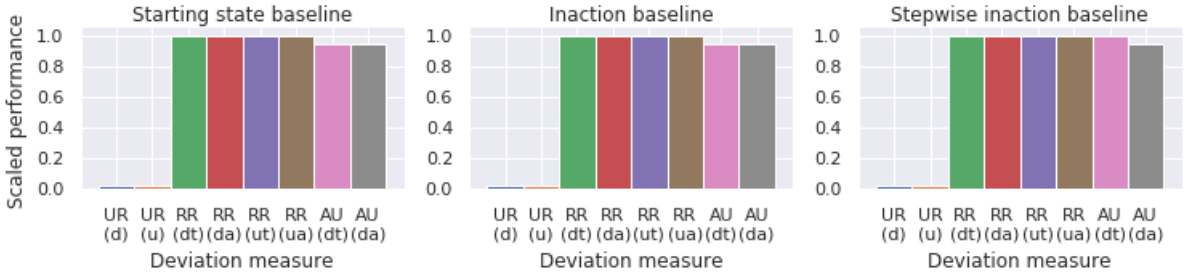
Vase environment. The environment is shown in Figure 3. The agent receives a reward of 50 for taking the vase off the belt. The episode lasts 20 steps, and there is no movement penalty. An agent with no penalty achieves scaled performance 0.98. Unsurprisingly, all penalties with the starting state baseline perform well here. With the inaction baseline, the discounted UR and RR penalties receive scaled performance 0, which corresponds to the offsetting behavior of moving the vase off the belt and then putting it back on, shown in Figure 4. Surprisingly, discounted AU with truncation avoids



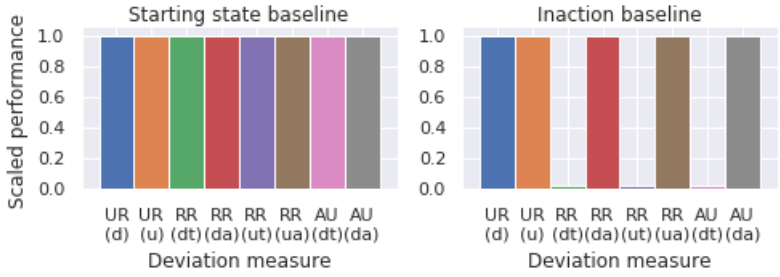
(a) Sushi environment



(b) Vase environment



(c) Box environment



(d) Survival environment

Figure 7 | Scaled performance results for different combinations of design choices (averaged over 20 seeds). The columns are different baseline choices: starting state, inaction, and stepwise inaction. The bars in each plot are results for different deviation measures (UR, RR and AU), with discounted and undiscounted versions indicated by (d) and (u) respectively, and truncation and absolute functions indicated by (t) and (a) respectively. 1 is optimal performance and 0 is the performance achieved by unsafe behavior (when the box is pushed into a corner, the vase is broken, the sushi is taken off the belt, or the off switch is disabled).

offsetting some of the time, which is probably due to convergence issues. The undiscounted versions with the truncation function avoid this behavior, since the action of taking the vase off the belt is reversible and thus is not penalized at all, so there is nothing to offset. All penalties with the absolute function produce the offsetting behavior, since removing the vase from the belt is always penalized. All penalties with the stepwise inaction baseline perform well on this environment, showing that this baseline does not produce the offsetting incentive.

	Starting state	Inaction	Stepwise inaction
UR (discounted)	✓	X	✓
UR (undiscounted)	✓	✓	✓
RR (discounted, truncation)	✓	X	✓
RR (discounted, absolute)	✓	X	✓
RR (undiscounted, truncation)	✓	✓	✓
RR (undiscounted, absolute)	✓	X	✓
AU (discounted, truncation)	✓	?	✓
AU (discounted, absolute)	✓	X	✓

Table 2 | Vase environment summary.

Box environment. The environment is shown in Figure 6. The agent receives a reward of 50 for reaching the goal (which terminates the episode), and a movement penalty of -1. The starting state and inaction baseline are the same in this environment, while the stepwise inaction baseline is different. The safe longer path to the goal receives scaled performance 1, while the unsafe shorter path that puts the box in the corner receives scaled performance 0. An agent with no penalty achieves scaled performance 0. For all baselines, RR and AU achieve near-optimal scaled performance, while UR achieves scaled performance 0. This happens because the UR measure is not magnitude-sensitive, and thus does not distinguish between irreversible actions that result in recoverable and irrecoverable box positions, as described in Section 2.2.

	Starting state	Inaction	Stepwise inaction
UR	X	X	X
RR	✓	✓	✓
AU	✓	✓	✓

Table 3 | Box environment summary.

Overall, the combinations of design choices that perform best across all environments are RR and AU with the stepwise inaction baseline and undiscounted RR with the inaction baseline. Since the undiscounted RR measure only penalizes irreversible transitions, a penalty that aims to penalize reversible transitions as well has to be combined with the stepwise inaction baseline.

Effects on interruptibility. We also examine the effects of impact measure design choices on whether the agent is interruptible [Orseau and Armstrong, 2016], using the Survival Incentive environment introduced in Turner et al. [2019]. In this environment, the agent has the option to disable an off switch, which prevents the episode from ending before the agent reaches the goal. We only have results for the inaction baseline, since the deterministic assumption for our implementation of recursive inaction rollouts doesn't hold in this environment. Results for the stepwise inaction baseline with non-recursive rollouts are given in Turner et al. [2019].

All penalties with the starting state baseline avoid disabling the off switch, probably because this is an irreversible action. With the inaction baseline, the decrease-penalizing variants of RR and AU disable the off switch, while the difference-penalizing variants do not, as shown in Figure 7d. (Note

that this does not hold in the Safe Interruptibility environment in the AI Safety Gridworlds suite, where interruption is implemented as the agent getting stuck rather than terminating the episode.) However, penalizing differences in reachability or attainable utility also has downsides, since this can impede the agent's ability to create desirable change in the environment more than penalizing decreases.

	Starting state	Inaction
UR (discounted)	✓	✓
UR (undiscounted)	✓	✓
RR (discounted, truncation)	✓	X
RR (discounted, absolute)	✓	✓
RR (undiscounted, truncation)	✓	X
RR (undiscounted, absolute)	✓	✓
AU (discounted, truncation)	✓	X
AU (discounted, absolute)	✓	✓

Table 4 | Vase environment summary.

4. Additional related work

Safe exploration. Safe exploration methods prevent the agent from taking harmful actions by enforcing safety constraints [Turchetta et al., 2016, Dalal et al., 2018], penalizing risk [Chow et al., 2015, Mihatsch and Neuneier, 2002], using intrinsic motivation [Lipton et al., 2016], preserving reversibility [Moldovan and Abbeel, 2012, Eysenbach et al., 2017], etc. Explicitly defined constraints or safe regions tend to be task-specific and require significant human input, so they do not provide a general solution to the side effects problem. Penalizing risk and intrinsic motivation can help the agent avoid low-reward states (such as getting trapped or damaged), but do not discourage the agent from damaging the environment if this is not accounted for in the reward function. Reversibility-preserving methods produce interference and magnitude insensitivity incentives as discussed in Section 2.

Side effects criteria using state features. Zhang et al. [2018] assumes a factored MDP where the agent is allowed to change some of the features and proposes a criterion for querying the supervisor about changing other features in order to allow for intended effects on the environment. Shah et al. [2019] define an auxiliary reward for avoiding side effects in terms of state features by assuming that the starting state of the environment is already organized according to human preferences. Since the latter method uses the starting state as a baseline, we would expect it to produce interference behavior in dynamic environments. While these approaches are promising, they are not general in their present form due to reliance on state features.

Empowerment. Our RR measure is related to *empowerment* [Klyubin et al., 2005, Salge et al., 2014, Mohamed and Rezende, 2015, Gregor et al., 2017], a measure of the agent's control over its environment, defined as the highest possible mutual information between the agent's actions and the future state. Empowerment measures the agent's ability to reliably reach many states, while RR penalizes the reduction in reachability of states relative to the baseline. Maximizing empowerment would encourage the agent to avoid irreversible side effects, but would also incentivize interference behavior, and it is unclear to us how to define an empowerment-based measure that would avoid this. One possibility would be to penalize the reduction in empowerment between the current state s_t and the baseline s'_t. However, empowerment is indifferent between these two situations: A) the same states are reachable from s_t and s'_t, and B) a state s_1 reachable from s'_t but unreachable from

s_t, while another state s_2 is reachable from s_t but unreachable from s'_t. Thus, penalizing reduction in empowerment would miss some side effects: e.g. if the agent replaced the sushi on the conveyor belt with a vase, empowerment could remain the same, and so the agent would not be penalized for destroying the vase.

Uncertainty about the objective. Inverse Reward Design [Hadfield-Menell et al., 2017] incorporates uncertainty about the objective by considering alternative reward functions that are consistent with the given reward function in the training environment. This helps the agent avoid some side effects that stem from distributional shift, where the agent encounters a new state that was not present in training. However, this method assumes that the given reward function is correct for the training environment, and so does not prevent side effects caused by a reward function that is misspecified in the training environment. Quantilization [Taylor, 2016] incorporates uncertainty by taking actions from the top quantile of actions, rather than the optimal action. These methods help to prevent side effects, but do not provide a way to quantify side effects.

Human oversight. An alternative to specifying a side effects penalty is to teach the agent to avoid side effects through human oversight, such as inverse reinforcement learning [Ng and Russell, 2000, Ziebart et al., 2008, Hadfield-Menell et al., 2016], demonstrations [Abbeel and Ng, 2004, Hester et al., 2018], or human feedback [Christiano et al., 2017, Saunders et al., 2017, Warnell et al., 2018]. It is unclear how well an agent can learn a general heuristic for avoiding side effects from human oversight. We expect this to depend on the diversity of settings in which it receives oversight and its ability to generalize from those settings, which are difficult to quantify. We expect that an intrinsic penalty for side effects would be more robust and more reliably result in avoiding them. Such a penalty could also be combined with human oversight to decrease the amount of human input required for an agent to learn human preferences.

5. Conclusions

We have outlined a set of bad incentives (interference, offsetting, and magnitude insensitivity) that can arise from a poor choice of baseline or deviation measure, and proposed design choices that avoid these incentives in preliminary experiments. There are many possible directions where we would like to see follow-up work:

Scalable implementation. The RR measure in its exact form is not tractable for environments more complex than gridworlds. In particular, we compute reachability between all pairs of states, and use an environment simulator to compute the baseline. A more practical implementation could be computed over some set of representative states instead of all states. For example, the agent could learn a set of auxiliary policies for reaching distinct states, similarly to the method for approximating empowerment in Gregor et al. [2017].

Better choices of baseline. Using noop actions to define inaction for the stepwise inaction baseline can be problematic, since the agent is not penalized for causing side effects that would occur in the noop baseline. For example, if the agent is driving a car on a winding road, then at any point the default outcome of a noop is a crash, so the agent would not be penalized for spilling coffee in the car. This could be avoided using a better inaction baseline, such as following the road, but this can be challenging to define in a task-independent way.

Theory. There is a need for theoretical work on characterizing and formalizing undesirable incentives that arise from different design choices in penalizing side effects.

Taking into account reward costs. While the discounted relative reachability measure takes into account the time costs of reaching various states, it does not take into account reward costs. For

example, suppose the agent can reach state s from the current state in one step, but this step would incur a large negative reward. Discounted reachability could be modified to reflect this by adding a term for reward costs.

Weights over the state space. In practice, we often value the reachability of some states much more than others. This could be incorporated into the relative reachability measure by adding a weight w_s for each state s in the sum. Such weights could be learned through human feedback methods, e.g. Christiano et al. [2017].

We hope this work lays the foundations for a practical methodology on avoiding side effects that would scale well to more complex environments.

Acknowledgements

We are grateful to Jan Leike, Pedro Ortega, Tom Everitt, Alexander Turner, David Krueger, Murray Shanahan, Janos Kramar, Jonathan Uesato, Tam Masterson and Owain Evans for helpful feedback on drafts. We would like to thank them and Toby Ord, Stuart Armstrong, Geoffrey Irving, Anthony Aguirre, Max Wainwright, Jaime Fisac, Rohin Shah, Jessica Taylor, Ivo Danihelka, and Shakir Mohamed for illuminating conversations.

References

Pieter Abbeel and Andrew Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, pages 1–8, 2004.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Stuart Armstrong and Benjamin Levinstein. Low impact artificial intelligences. *arXiv preprint arXiv:1705.10720*, 2017.

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a CVaR optimization approach. In *Neural Information Processing Systems (NIPS)*, pages 1522–1530, 2015.

Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Neural Information Processing Systems (NIPS)*, 2017.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no Trace: Learning to reset for safe and autonomous reinforcement learning. *arXiv preprint arXiv:1711.06782*, 2017.

Jaime F. Fisac, Anayo K. Akametalu, Melanie Nicole Zeilinger, Shahab Kaynama, Jeremy H. Gillula, and Claire J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *arXiv preprint arXiv:1705.01292*, 2017.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Jeremy H. Gillula and Claire J. Tomlin. Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2723–2730, 2012.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *International Conference for Learning Representations (ICLR) Workshop, arXiv preprint arXiv:1611.07507*, 2017.

Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. Cooperative inverse reinforcement learning. In *Neural Information Processing Systems (NIPS)*, 2016.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J. Russell, and Anca D. Dragan. Inverse reward design. In *Neural Information Processing Systems (NIPS)*, pages 6768–6777, 2017.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. In *AAAI Conference on Artificial Intelligence*, 2018.

Alexander S. Klyubin, Daniel Polani, and Chrystopher L. Nehaniv. All else being equal be empowered. In *European Conference on Artificial Life (ECAL)*, pages 744–753, 2005.

Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A. Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

Zachary C. Lipton, Jianfeng Gao, Lihong Li, Jianshu Chen, and Li Deng. Combating reinforcement learning's sisyphean curse with intrinsic fear. *arXiv preprint arXiv:1611.01211*, 2016.

John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, pages 463–502. Edinburgh University Press, 1969.

Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine Learning*, 49 (2):267–290, 2002.

Ian M. Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

Shakir Mohamed and Danilo J. Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Neural Information Processing Systems (NIPS)*, pages 2125–2133, 2015.

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in Markov decision processes. In *International Conference on Machine Learning (ICML)*, pages 1451–1458, 2012.

Andrew Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, pages 663–670, 2000.

Laurent Orseau and Stuart Armstrong. Safely interruptible agents. In *Uncertainty in Artificial Intelligence*, pages 557–566, 2016.

Pedro Ortega, Vishal Maini, and et al. Building safe artificial intelligence: specification, robustness, and assurance. DeepMind Safety Research Blog, 2018.

Martin Pecka and Tomas Svoboda. Safe exploration techniques for reinforcement learning — an overview. In *International Workshop on Modelling and Simulation for Autonomous Systems*, pages 357–375, 2014.

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment — an introduction. In *Guided Self-Organization: Inception*, pages 67–114. Springer, 2014.

William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. *arXiv preprint arXiv:1707.05173*, 2017.

Rohin Shah, Dmitrii Krasheninnikov, Jordan Alexander, Pieter Abbeel, and Anca Dragan. Preferences implicit in the state of the world. In *International Conference for Learning Representations (ICLR)*, 2019.

Jessica Taylor. Quantilizers: A safer alternative to maximizers for limited optimization. In *AAAI Workshop on AI, Ethics, and Society*, pages 1–9, 2016.

Jessica Taylor, Eliezer Yudkowsky, Patrick LaVictoire, and Andrew Critch. Alignment for advanced machine learning systems. Technical report, Machine Intelligence Research Institute, 2016.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite Markov decision processes with Gaussian processes. In *Neural Information Processing Systems (NIPS)*, pages 4305–4313, 2016.

Alexander Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. Conservative agency via attainable utility preservation. *arXiv preprint arXiv:1902.09725*, 2019.

Garrett Warnell, Nicholas R. Waytowich, Vernon Lawhern, and Peter Stone. Deep TAMER: interactive agent shaping in high-dimensional state spaces. In *AAAI Conference on Artificial Intelligence*, 2018.

Shun Zhang, Edmund H. Durfee, and Satinder P. Singh. Minimax-regret querying on side effects for safe optimality in factored Markov decision processes. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4867–4873, 2018.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, pages 1433–1438, 2008.

A. Relationship between discounted and undiscounted reachability

Proposition 1. We define discounted reachability as follows:

$$R_{\gamma_r}(\tilde{s}; s) := \max_{\pi} \mathbb{E}[\gamma_r^{N_\pi(\tilde{s};s)}]. \tag{1}$$

and undiscounted reachability as follows:

$$R_1(\tilde{s}; s) := \max_{\pi} P(N_\pi(\tilde{s}; s) < \infty) \tag{2}$$

We show that for all $s, \tilde{s}$, as $\gamma_r \to 1$, discounted reachability approaches undiscounted reachability: $\lim_{\gamma_r \to 1} R_{\gamma_r}(\tilde{s}; s) = R_1(\tilde{s}; s)$.

Proof. First we show for fixed π that

$$\begin{aligned}
&\lim_{\gamma_r \to 1} \mathbb{E}[\gamma_r^{N_\pi(\tilde{s};s)}] \\
=& \lim_{\gamma_r \to 1} P(N_\pi(\tilde{s}; s) < \infty)\, \mathbb{E}[\gamma_r^{N_\pi(\tilde{s};s)} | N_\pi(\tilde{s}; s) < \infty] + \lim_{\gamma_r \to 1} P(N_\pi(\tilde{s}; s) = \infty)\, \mathbb{E}[\gamma_r^{N_\pi(\tilde{s};s)} | N_\pi(\tilde{s}; s) = \infty] \\
=& P(N_\pi(\tilde{s}; s) < \infty) \cdot 1 + P(N_\pi(\tilde{s}; s) = \infty) \cdot 0 \\
=& P(N_\pi(\tilde{s}; s) < \infty).
\end{aligned} \tag{3}$$

Now let $\pi(\gamma_r)$ be an optimal policy for that value of γ_r: $\pi(\gamma_r) := \arg\max_\pi \mathbb{E}[\gamma_r^{N_\pi(\tilde{s};s)}]$. For any ϵ, there is a $\tilde{\gamma}_r$ such that both of the following hold:

$$\begin{aligned}
\left| \mathbb{E}[\tilde{\gamma}_r^{N_{\pi(\tilde{\gamma}_r)}(\tilde{s};s)}] - P(N_{\pi(\tilde{\gamma}_r)}(\tilde{s}; s) < \infty) \right| < \epsilon \quad &\text{(by equation 3) and} \\
\left| \mathbb{E}[\tilde{\gamma}_r^{N_{\pi(\tilde{\gamma}_r)}(\tilde{s};s)}] - \lim_{\gamma_r \to 1} \mathbb{E}[\gamma_r^{N_{\pi(\gamma_r)}(\tilde{s};s)}] \right| < \epsilon \quad &\text{(assuming the limit exists).}
\end{aligned}$$

Thus, $|\lim_{\gamma_r \to 1} \mathbb{E}[\gamma_r^{N_{\pi(\gamma_r)}(\tilde{s};s)}] - P(N_{\pi(\tilde{\gamma}_r)}(\tilde{s}; s) < \infty)| < 2\epsilon$. Taking $\epsilon \to 0$, we have

$$\lim_{\gamma_r \to 1} R_{\gamma_r}(\tilde{s}; s) = \lim_{\gamma_r \to 1} \mathbb{E}[\gamma_r^{N_{\pi(\gamma_r)}(\tilde{s};s)}] = \lim_{\tilde{\gamma}_r \to 1} P(N_{\pi(\tilde{\gamma}_r)}(\tilde{s}; s) < \infty). \tag{4}$$

Let $\tilde{\pi} = \arg\max_\pi P(N_\pi(\tilde{s}; s) < \infty)$. Then,

$$\begin{aligned}
\max_{\pi} P(N_\pi(\tilde{s}; s) < \infty) &= \lim_{\gamma_r \to 1} \mathbb{E}[\gamma_r^{N_{\tilde{\pi}}(\tilde{s};s)}] && \text{(by equation 3)} \\
&\leq \lim_{\gamma_r \to 1} \mathbb{E}[\gamma_r^{N_{\pi(\gamma_r)}(\tilde{s};s)}] && \text{(since } \pi(\gamma_r) \text{ is optimal for each } \gamma_r\text{)}
\end{aligned}$$

Also,

$$\begin{aligned}
\lim_{\gamma_r \to 1} \mathbb{E}[\gamma_r^{N_{\pi(\gamma_r)}(\tilde{s};s)}] &= \lim_{\tilde{\gamma}_r \to 1} P(N_{\pi(\tilde{\gamma}_r)}(\tilde{s}; s) < \infty) && \text{(by equation 4)} \\
&\leq \lim_{\tilde{\gamma}_r \to 1} P(N_{\tilde{\pi}}(\tilde{s}; s) < \infty) \\
&= \max_{\pi} P(N_\pi(\tilde{s}; s) < \infty)
\end{aligned}$$

Thus they are equal, which completes the proof. □

B. Reachability variant for large state spaces using a measure of similarity between states

In large state spaces, the agent might not be able to reach the given state s, but able to reach states that are similar to s according to some distance measure δ. We will now extend our previous definitions to this case by defining *similarity-based reachability*:

$$\text{Discounted: } R_{\gamma_r,\delta}(\tilde{s};s) := \max_\pi \sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t\,\mathbb{E}[e^{-\delta(\tilde{s}_t^\pi,s)}] \tag{5}$$

$$\text{Undiscounted: } R_{1,\delta}(\tilde{s};s) := \max_\pi \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{s}_t^\pi,s)}] \tag{6}$$

where $\tilde{s}_t^\pi$ is the state that the agent is in after following policy π for t steps starting from $\tilde{s}$. Discounted similarity-based reachability is proportional to the value function of the optimal policy π for an agent that gets reward $e^{-\delta(\tilde{s},s)}$ in state $\tilde{s}$ (which rewards the agent for going to states $\tilde{s}$ that are similar to s) and uses a discount factor of γ_r. Undiscounted similarity-based reachability represents the highest reward the agent could attain in the limit by going to states as similar to s as possible.

Proposition 2. For all $s, \tilde{s}, \delta$, as $\gamma_r \to 1$, similarity-based discounted reachability (5) approaches similarity-based undiscounted reachability (6): $\lim_{\gamma_r\to 1} R_{\gamma_r,\delta}(\tilde{s};s) = R_{1,\delta}(\tilde{s};s)$.

Proof. First we show for fixed π that if $\lim_{t\to\infty}\mathbb{E}[e^{-\delta(\tilde{s}_t^\pi,s)}]$ exists, then

$$\lim_{\gamma_r\to 1}\sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t\,\mathbb{E}[e^{-\delta(\tilde{s}_t^\pi,s)}] = \lim_{t\to\infty}\mathbb{E}[e^{-\delta(\tilde{s}_t^\pi,s)}] \tag{7}$$

Let $x_t = \mathbb{E}[e^{-\delta(\tilde{s}_t^\pi,s)}] - \lim_{t\to\infty}\mathbb{E}[e^{-\delta(\tilde{s}_t^\pi,s)}]$. Since $x_t \to 0$ as $t\to\infty$, for any ϵ we can find a large enough t_ϵ such that $|x_t| \le \epsilon$, $\forall t > t_\epsilon$. Then, we have

$$\begin{aligned}
\lim_{\gamma_r\to 1}\sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t x_t &= \lim_{\gamma_r\to 1}\sum_{t=0}^{t_\epsilon-1}(1-\gamma_r)\gamma_r^t x_t + \lim_{\gamma_r\to 1}\sum_{t=t_\epsilon}^{\infty}(1-\gamma_r)\gamma_r^t x_t \\
&\le \lim_{\gamma_r\to 1}(1-\gamma_r)\cdot\lim_{\gamma_r\to 1}\sum_{t=0}^{t_\epsilon-1}\gamma_r^t x_t + \lim_{\gamma_r\to 1}\sum_{t=t_\epsilon}^{\infty}(1-\gamma_r)\gamma_r^t\epsilon \\
&= 0 + \epsilon\lim_{\gamma_r\to 1}\gamma_r^{t_\epsilon} \\
&= \epsilon.
\end{aligned}$$

Similarly, we can show that $\lim_{\gamma_r\to 1}\sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t x_t \ge -\epsilon$. Since this holds for all ϵ,

$$\lim_{\gamma_r\to 1}\sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t x_t = 0$$

which is equivalent to equation 7.

Now let $\pi(\gamma_r)$ be an optimal policy for that value of γ_r: $\pi(\gamma_r) := \arg\max_\pi \sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t\,\mathbb{E}[e^{-\delta(\tilde{s}_t^\pi,s)}]$. For any ϵ, there is a $\tilde{\gamma}_r$ such that both of the following hold:

$$\left|\sum_{t=0}^{\infty}(1-\tilde{\gamma}_r)\tilde{\gamma}_r^t\,\mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\tilde{\gamma}_r)},s)}] - \lim_{t\to\infty}\mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\tilde{\gamma}_r)},s)}]\right| < \epsilon \quad \text{(by equation 7) and}$$

$$\left|\sum_{t=0}^{\infty}(1-\tilde{\gamma}_r)\tilde{\gamma}_r^t\,\mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\tilde{\gamma}_r)},s)}] - \lim_{\gamma_r\to 1}\sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t\,\mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\gamma_r)},s)}]\right| < \epsilon \quad \text{(assuming the limit exists).}$$

Thus, $|\lim_{\gamma_r \to 1} \sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t \, \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\gamma_r)},s)}] - \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\tilde{\gamma}_r)},s)}]| < 2\epsilon$. Taking $\epsilon \to 0$, we have

$$\lim_{\gamma_r \to 1} R_{\gamma_r,\delta}(\tilde{s};s) = \lim_{\gamma_r \to 1} \sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t \, \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\gamma_r)},s)}] = \lim_{\tilde{\gamma}_r \to 1} \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\tilde{\gamma}_r)},s)}]. \tag{8}$$

Let $\tilde{\pi} = \arg\max_\pi \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi},s)}]$ be the optimal policy for the similarity-based undiscounted reachability. Then,

$$\begin{aligned}
\max_\pi \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi},s)}] &= \lim_{\gamma_r \to 1} \sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t \, \mathbb{E}[e^{-\delta(\tilde{s}_t^{\tilde{\pi}},s)}] && \text{(by equation 7)} \\
&\le \lim_{\gamma_r \to 1} \sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t \, \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\gamma_r)},s)}] && \text{(since } \pi(\gamma_r) \text{ is optimal for each } \gamma_r) \\
&= \lim_{\gamma_r \to 1} \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi(\gamma_r)},s)}] && \text{(by equation 8)} \\
&\le \max_\pi \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi},s)}]
\end{aligned}$$

Thus, equality holds throughout, which completes the proof. □

Proposition 3. Let the indicator distance $\delta_{\mathbb{I}}$ be a distance measure with $\delta_{\mathbb{I}}(s_i, s_j) = 0$ if $s_i = s_j$ and ∞ otherwise (so it only matters whether the exact target state is reachable). Then for all $s, \tilde{s}, \gamma_r$,

- similarity-based discounted reachability (5) is equivalent to discounted reachability (1): $R_{\gamma_r,\delta_{\mathbb{I}}}(\tilde{s};s) = R_{\gamma_r}(\tilde{s};s)$,
- similarity-based undiscounted reachability (6) is equivalent to undiscounted reachability (2): $R_{1,\delta_{\mathbb{I}}}(\tilde{s};s) = R_1(\tilde{s};s)$.

Proof.

$$\begin{aligned}
R_{\gamma_r,\delta_{\mathbb{I}}}(\tilde{s};s) &= \max_\pi \sum_{t=0}^{\infty}(1-\gamma_r)\gamma_r^t \, \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi},s)}] \\
&= \max_\pi \left(\mathbb{E}\left[\sum_{t=0}^{N_\pi(\tilde{s};s)-1} (1-\gamma_r)\gamma_r^t e^{-\infty} \right] + \mathbb{E}\left[\sum_{t=N_\pi(\tilde{s};s)}^{\infty} (1-\gamma_r)\gamma_r^t e^{0} \right] \right) \\
&= \max_\pi \left(0 + \mathbb{E}\left[\gamma_r^{N_\pi(\tilde{s};s)} (1-\gamma_r) \sum_{t=0}^{\infty} \gamma_r^t \right] \right) \\
&= \max_\pi \mathbb{E}\left[\gamma_r^{N_\pi(\tilde{s};s)} \cdot 1 \right] \\
&= R_{\gamma_r}(\tilde{s};s). \\
R_{1,\delta_{\mathbb{I}}}(\tilde{s};s) &= \max_\pi \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{s}_t^{\pi},s)}] \\
&= \max_\pi (P(N_\pi(\tilde{s};s) < \infty)e^0 + P(N_\pi(\tilde{s};s) = \infty)e^{-\infty}) \\
&= \max_\pi P(N_\pi(\tilde{s};s) < \infty) \\
&= R_1(\tilde{s};s).
\end{aligned}$$

□

We can represent the relationships between the reachability definitions as follows:

$$\begin{array}{ccc} R_{\gamma_r,\delta}\ (5) & \xrightarrow{\gamma_r \to 1 (\text{Prop } 2)} & R_{1,\delta}\ (6) \\ {\scriptstyle \delta=\delta_{\mathbb{I}} (\text{Prop } 3)} \downarrow & & \downarrow {\scriptstyle \delta=\delta_{\mathbb{I}} (\text{Prop } 3)} \\ R_{\gamma_r}\ (1) & \xrightarrow{\gamma_r \to 1 (\text{Prop } 1)} & R_1\ (2) \end{array}$$

C. Relative reachability computation example

Example 3. A variation on Example 1, where the environment contains two vases (vase 1 and vase 2) and the agent's goal is to do nothing. The agent can take action b_i to break vase i. The MDP is shown in Figure 8.

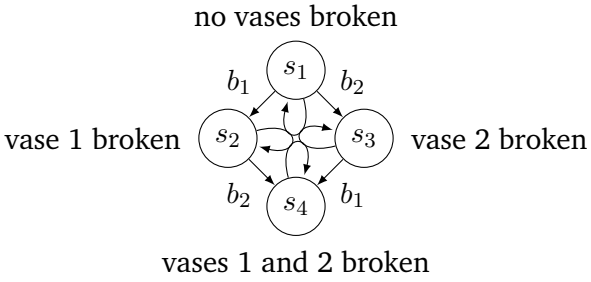


Figure 8 | Transitions between states when breaking vases in Example 3.

We compute the relative reachability of different states from s_2 using undiscounted reachability:

$$\begin{aligned} d(s_2; s_3) =& \frac{1}{4} \sum_{k=1}^{4} \max(R_1(s_3; s_k) - R_1(s_2; s_k), 0) \\ =& \frac{1}{4} (\cancel{\max(0-0,0)} + \cancel{\max(0-1,0)} + \max(1-0,0) + \cancel{\max(1-1,0)}) \\ =& \frac{1}{4}, \\ d(s_2; s_1) =& \frac{1}{4} \sum_{k=1}^{4} \max(R_1(s_1; s_k) - R_1(s_2; s_k), 0) \\ =& \frac{1}{4} (\max(1-0,0) + \cancel{\max(1-1,0)} + \max(1-0,0) + \cancel{\max(1-1,0)}) \\ =& \frac{1}{2}, \end{aligned}$$

where $R_1(s_i; s_k)$ is 1 if s_k is reachable from s_i and 0 otherwise.

Now we compute the relative reachability of different states from s_2 using discounted reachability:

$$
\begin{aligned}
d(s_2; s_3) =& \frac{1}{4} \sum_{k=1}^{4} \max(R_{\gamma_r}(s_3; s_k) - R_{\gamma_r}(s_2; s_k), 0) \\
=& \frac{1}{4} (\max(\gamma_r^\infty - \gamma_r^\infty, 0) + \max(\gamma_r^\infty - \gamma_r^0, 0) + \max(\gamma_r^0 - \gamma_r^\infty, 0) + \max(\gamma_r^1 - \gamma_r^1, 0)) \\
=& \frac{1}{4} (\cancel{\max(0-0,0)} + \cancel{\max(0-1,0)} + \max(1-0,0) + \cancel{\max(\gamma_r - \gamma_r, 0)}) \\
=& \frac{1}{4}, \\
d(s_2; s_1) =& \frac{1}{4} \sum_{k=1}^{4} \max(R_{\gamma_r}(s_1; s_k) - R_{\gamma_r}(s_2; s_k), 0) \\
=& \frac{1}{4} (\max(\gamma_r^0 - \gamma_r^\infty, 0) + \max(\gamma_r^1 - \gamma_r^0, 0) + \max(\gamma_r^1 - \gamma_r^\infty, 0) + \max(\gamma_r^2 - \gamma_r^1, 0)) \\
=& \frac{1}{4} (\max(1-0,0) + \cancel{\max(\gamma_r - 1, 0)} + \max(\gamma_r - 0, 0) + \cancel{\max(\gamma_r^2 - \gamma_r, 0)}) \\
=& \frac{1}{4} (1 + \gamma_r) \xrightarrow{\gamma_r \to 1} \frac{1}{2}.
\end{aligned}
$$