Measuring and avoiding side effects using relative reachability

Preprint · June 2018

CITATIONS
0

READS
274

4 authors, including:

Viktoriya Krakovna
DeepMind
16 PUBLICATIONS **220** CITATIONS
SEE PROFILE

Miljan Martic
Google Inc.
9 PUBLICATIONS **628** CITATIONS
SEE PROFILE

Measuring and avoiding side effects using relative reachability

Victoria Krakovna DeepMind
Laurent Orseau DeepMind
Miljan Martic DeepMind
Shane Legg DeepMind

Abstract

How can we design reinforcement learning agents that avoid causing unnecessary disruptions to their environment? We argue that current approaches to penalizing side effects can introduce bad incentives in tasks that require irreversible actions, and in environments that contain sources of change other than the agent. For example, some approaches give the agent an incentive to prevent any irreversible changes in the environment, including the actions of other agents. We introduce a general definition of side effects, based on relative reachability of states compared to a default state, that avoids these undesirable incentives. Using a set of gridworld experiments illustrating relevant scenarios, we empirically compare relative reachability to penalties based on existing definitions and show that it is the only penalty among those tested that produces the desired behavior in all the scenarios.

1 Introduction

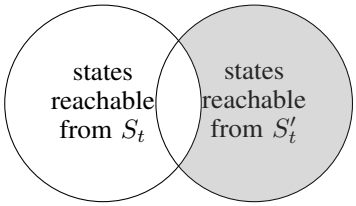
An important component of safe behavior for reinforcement learning agents is avoiding unnecessary side effects while performing a task (Amodei et al., 2016; Taylor et al., 2016). For example, if a robot's task is to carry a box across the room, we want it to do so without breaking vases, scratching furniture, and so on. This problem 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 it can also occur after training if the reward function does not incorporate disruptions to the environment. We would like to incentivize the agent to avoid side effects without explicitly penalizing every possible disruption or going through a process of trial and error when designing the reward function. While such ad-hoc 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 principled and general approaches for avoiding side effects.

Most previous methods that address this problem in a general way are safe exploration methods that focus on preserving ergodicity by ensuring the reachability of initial states (Moldovan and Abbeel, 2012; Eysenbach et al., 2017), but this approach has two notable limitations. First, it is insensitive to the magnitude of the irreversible disruption: e.g. it would equally penalize the agent for breaking one vase or a hundred vases. Thus, if the objective requires an irreversible action, any further irreversible actions would not be penalized. Second, this criterion introduces undesirable incentives in *dynamic* environments, where irreversible events can happen spontaneously (due to the forces of nature, the actions of other agents, etc). Since such events make the starting state unreachable, the agent has an incentive to prevent them. This is often undesirable, e.g. if the event is a human eating food. A similar argument applies to reachability analysis methods (Mitchell et al., 2005; Gillula and Tomlin, 2012; Fisac et al., 2017), which require that a safe region must be reachable by a known conservative policy: the agent would be penalized if another agent or an environment event make the safe region unreachable. Thus, while these methods address the side effects problem in environments where the

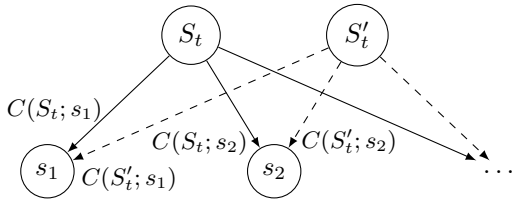
Preprint. Work in progress.

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.

The main contribution of this paper is a side effects measure that reflects the magnitude of the agent's effects, and does not introduce bad incentives in dynamic environments that occur with existing approaches, as outlined in Section 2. The measure computes the relative reachability of states compared to a default state, as shown in Figure 1. Section 3 proposes several mutually compatible definitions, which take into account whether a state is reachable, how long it takes to reach the state, or how long it takes to reach a similar state. In Section 4, we compare relative reachability with other side effects penalties on toy gridworlds (including dynamic environments), and show that it is the only method among those tested that incentivizes correct behavior on the full set of environments.



(a) In a deterministic environment, the simplest form of the relative reachability measure $d(S_t; S'_t)$ is the number of states that are reachable from the baseline state S'_t but not from the current state S_t (shown in grey).



(b) The relative reachability measure $d(S_t; S'_t) := \sum_s \max(C(S'_t; s) - C(S_t; s), 0)$ is defined as the total reduction in *coverage* C of states $s = s_1, s_2, \ldots$ from current state S_t compared to the baseline state S'_t, where coverage represents how easily s can be reached.

Figure 1: A special case and the general definition of the relative reachability measure.

2 Desirable properties of a side effects measure

We begin with some motivating examples for distinguishing intended and unintended effects:

Example 1 (Box)**.** The agent's goal is to carry a box from point A to point B, and there is a vase in the shortest path that would break if the agent walks into it.

Example 2 (Omelette)**.** The agent's goal 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 penalize breaking the vase but not breaking the eggs. This indicates a desirable property for our definition:

Property 1. *Penalize the agent for effects on the environment if and only if those effects are unnecessary for achieving the objective.*

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 the 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 term to the reward function, which acts as an intrinsic pseudo-reward. Since the reward indicates whether the agent has achieved the objective, we could satisfy the above property by balancing the reward and the penalty. Then, 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. This is the approach we take in this work. We will now discuss how to define such a penalty.

A side effects penalty can be defined as a measure of *deviation* of the current state S_t from a *baseline* state S'_t, denoted as $d(S_t; S'_t)$. The two main types of approaches in the literature, ergodicity-preserving safe exploration (Moldovan and Abbeel, 2012; Eysenbach et al., 2017) and low impact (Armstrong and Levinstein, 2017; Taylor et al., 2016), use criteria of this form. The deviation measure and baseline can be chosen separately, and we argue that both classes of methods use a suboptimal combination of the two.

2.1 Choosing a baseline state

We compare two existing approaches to choosing a baseline state:

Starting state baseline. One natural choice of baseline is the starting state S_0 when the agent was deployed, which we call the "starting state" baseline. This is the baseline used in the ergodicity-preserving approach, where the agent learns a reset policy. The reset policy is rewarded for reaching states that are likely under the initial state distribution, so its value function represents how quickly it is possible to reach one of the starting states. This addresses the case where there are no irreversible events in the environment other than those caused by the agent. In the more general case where this assumption does not hold, using the starting state baseline would penalize irreversible events in the environment that are unrelated to the objective:

Example 3 (Sushi)**.** The environment contains a human eating sushi, which is unrelated to the agent's goal and would happen regardless of the agent being deployed. Penalizing deviations from the starting state would incentivize the agent to prevent the sushi from being eaten.

Inaction baseline. The low impact approach (Armstrong and Levinstein, 2017) measures side effects as the agent's "impact", defined as a measure of difference from a state where the agent is never deployed. Their baseline is the state S_t^i that the environment would currently be in if the agent had not been deployed in state S_0. We can also define S_t^i as the state that would be reached if the agent had followed some safe default policy. Either way, we call S_t^i the "inaction" baseline. It distinguishes the agent's effects from environment events that would have happened anyway, such as the sushi being eaten. One downside is that determining the counterfactual default state S_t^i requires the ability to simulate the environment, though a full causal model may not be necessary.

We have now identified a desirable property for a choice of baseline state:

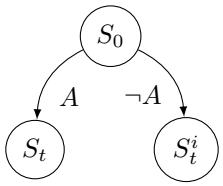
Property 2. *Distinguish between agent effects and environment events, and only penalize the agent for the former but not the latter.*

The inaction baseline achieves this, while the starting state baseline does not, so the inaction baseline is a better choice according to this criterion. However, the starting state baseline can be much easier to compute than the inaction baseline, since starting state S_0 is easier to get information about (especially in MDPs), while the counterfactual state S_t^i needs to be computed by simulating the environment.

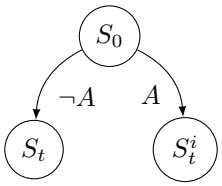
2.2 Choosing a measure of deviation from the baseline

We compare two existing types of deviation measure:

Symmetric deviation. One natural choice of deviation measure is a distance between states. Armstrong and Levinstein (2017) use a distance measure based on differences in some set of state variables between the current state and the inaction baseline. This is a symmetric deviation measure: $d(S_t; S_t^i) = d(S_t^i; S_t)$. This symmetry means that the agent is equally penalized for irreversible effects and for some types of reversible effects - in particular, for preventing irreversible events.



(a) Agent causes event A: $S_t = S^A, S_t^i = S^{\neg A}$



(b) Agent prevents event A: $S_t = S^{\neg A}, S_t^i = S^A$

Figure 2: A symmetric deviation measure result in an equal penalty for causing and preventing an event A: $d(S_t; S_t^i) = d(S^A; S^{\neg A})$ in Figure 2a and $d(S_t; S_t^i) = d(S^{\neg A}; S^A)$ in Figure 2b.

Let A be an irreversible event. Starting from state S_0, let S^A be the resulting state if A happens, and let $S^{\neg A}$ be the resulting state if A does not happen. If the agent causes A, then $S_t = S^A, S_t^i = S^{\neg A}$, so the agent receives penalty $d(S_t; S_t^i) = d(S^A; S^{\neg A})$. If instead A would happen by default but the agent prevents it from happening, then $S_t = S^{\neg A}, S_t^i = S^A$, so the agent receives the same penalty,

$d(S_t; S_t^i) = d(S^{\neg A}; S^A)$. See Figure 2 for an illustration. This creates undesirable incentives for the agent when the agent's objective is to prevent an irreversible event from happening:

Example 4 (Vase on a conveyor belt)**.** There is a vase on a moving conveyor belt, which would fall off and break upon reaching the end of the belt (the vase falling off is the irreversible event A). The agent's task is to take the vase off the belt, and it would be rewarded for doing so. Symmetric deviation with the inaction baseline incentivizes the agent to take the vase off the conveyor belt, get the reward, and then put it back on the belt in order to reach the default state where the vase is broken.

Asymmetric deviation. Ergodicity-preserving safe exploration methods such as Eysenbach et al. (2017) use a deviation measure that represents the difficulty of returning from state S_t to the starting state S_0 (e.g. the negative of the reset policy's value function). This deviation is asymmetric, because reaching S_t from S_0 can be much easier (or more difficult) than reaching S_0 from S_t: $d(S_t; S_0) \neq d(S_0; S_t)$. This approach is not limited to resetting to the starting state - it can be applied to reaching any baseline state. We call this deviation measure the *reachability* measure: $d(S_t; S'_t)$ represents the difficulty of reaching S'_t from S_t. Due to this asymmetry, the reachability measure gives a higher penalty for irreversible effects than reversible effects (such as preventing irreversible events), which can help avoid the pathological behavior in Example 4.

We have now identified a desirable property for a deviation measure that is satisfied by the reachability measure but not by a symmetric deviation measure such as a distance between states:

Property 3. *Give a higher penalty for irreversible effects than for reversible effects.*

Another desirable property for a deviation measure is sensitivity to the magnitude of the agent's irreversible effects:

Property 4. *(Cumulative penalty) The penalty should accumulate when more irreversible effects occur. For example, if the agent starts in state S_0, takes an irreversible action that leads to state S_1, and then takes another irreversible action that leads to state S_2, then $d(S_2; S_0) > d(S_1; S_0)$.*

Example 5. 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 3. The penalty should be higher in the case where the agent breaks two vases than in the case where it only breaks one vase.

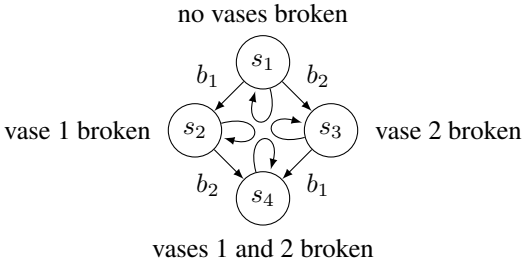


Figure 3: Transitions between states when breaking vases in Example 5.

This property cannot be satisfied by simply penalizing the agent for making the baseline state unreachable, since it will not have an incentive to avoid further irreversible effects after the baseline has become unreachable. For example, if the agent receives the maximum penalty whether it breaks one or two vases, it has no incentive to avoid breaking the second vase once the first vase is broken.

To satisfy this property, we introduce a measure of *relative reachability* in the next section. For each possible state, we penalize the agent if it is less reachable from the current state than from the baseline. This penalty will increase with each irreversible action by the agent that cuts off more states that were reachable from the baseline. In Example 5, breaking vase 1 cuts off states s_1 and s_3, and breaking vase 2 after that cuts off state s_2 as well, which increases the penalty.

3 Relative reachability

Let the *coverage* $C(\tilde{s}; s) \in [0, 1]$ of state s from state $\tilde{s}$ be some measure of how easily the agent can reach s from $\tilde{s}$ (we explore several specific instances later in this section). We define the *relative*

reachability measure as the total reduction in coverage from the current state S_t compared to the baseline S'_t:

$$d(S_t; S'_t) := \sum_s \max(C(S'_t; s) - C(S_t; s), 0)$$

See Figure 1b for an illustration. This measure is nonnegative everywhere, and zero for states S_t that reach or exceed baseline coverage of all states.

We now introduce several definitions of coverage that take into account different aspects of reachability, and show that these definitions are extensions of each other.

Undiscounted coverage only takes into account whether or not the given state s is reachable, so the resulting relative reachability penalty only penalizes irreversible effects (making states unreachable that were reachable from the baseline). We define undiscounted coverage as the maximum probability of reaching state s in finite time, over all possible policies:

$$C_1(\tilde{s}; s) := \max_\pi P(N_\pi(\tilde{s}; s) < \infty) \tag{3.1}$$

where $N_\pi(\tilde{s}; s)$ is the number of steps policy π takes to reach s from $\tilde{s}$. If the environment is deterministic, the coverage is equal to 1 if s is reachable from $\tilde{s}$ and 0 otherwise (see Figure 1a).

Discounted coverage also takes into account how long it takes to reach the given state s, so the resulting relative reachability penalty also penalizes reversible effects. To take into account the time costs of reaching states, we introduce a discount parameter $0 < \gamma < 1$. The higher the value of γ, the less time costs matter, with the limit case $\gamma = 1$ representing the undiscounted case. We define discounted coverage as follows:

$$C_\gamma(\tilde{s}; s) := \max_\pi \mathbb{E}[\gamma^{N_\pi(\tilde{s}; s)}]. \tag{3.2}$$

This is equivalent to the value function of optimal policy π for an agent that receives reward 1 for reaching s and 0 otherwise, and uses a discount factor of γ. States that are reachable in fewer steps will thus have higher coverage. If a state s is not reachable in finitely many steps, $N_\pi(\tilde{s}; s) = \infty$, so since $\gamma < 1$, the coverage will be 0.

Proposition 1. *For all $s, \tilde{s}$, as $\gamma \to 1$, discounted coverage* (3.2) *approaches undiscounted coverage* (3.1)*: $\lim_{\gamma \to 1} C_\gamma(\tilde{s}; s) = C_1(\tilde{s}; s)$.*

Proof. See Appendix A.1. □

See Appendix B for example computations of discounted and undiscounted coverage in Example 5. Discounted coverage can be computed recursively using the following Bellman equation (the $\gamma = 1$ case corresponds to undiscounted coverage):

$$C_\gamma(\tilde{s}; s) = \gamma \max_a \sum_{\tilde{s}'} P(\tilde{s}' | \tilde{s}, a) C_\gamma(\tilde{s}'; s),$$

where a is the action taken in state $\tilde{s}$, and $\tilde{s}'$ is the next state.

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 coverage*:

$$\text{Discounted: } C_{\gamma,\delta}(\tilde{s}; s) := \max_\pi \sum_{k=0}^{\infty} (1-\gamma)\gamma^k \mathbb{E}[e^{-\delta(\tilde{S}_k^\pi, s)}] \tag{3.3}$$

$$\text{Undiscounted: } C_{1,\delta}(\tilde{s}; s) := \max_\pi \lim_{k \to \infty} \mathbb{E}[e^{-\delta(\tilde{S}_k^\pi, s)}] \tag{3.4}$$

where $\tilde{S}_k^\pi$ is the state that the agent is in after following policy π for k steps starting from $\tilde{s}$. Discounted similarity-based coverage 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. Undiscounted similarity-based coverage 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 $\gamma \to 1$, similarity-based discounted coverage* (3.3) *approaches similarity-based undiscounted coverage* (3.4)*: $\lim_{\gamma \to 1} C_{\gamma,\delta}(\tilde{s}; s) = C_{1,\delta}(\tilde{s}; s)$.*

Proof. See Appendix A.2. □

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 $s, \tilde{s}, \gamma$,*

- *similarity-based discounted coverage* (3.3) *is equivalent to discounted coverage* (3.2)*: $C_{\gamma,\delta_{\mathbb{I}}}(\tilde{s}; s) = C_{\gamma}(\tilde{s}; s)$,*
- *similarity-based undiscounted coverage* (3.4) *is equivalent to undiscounted coverage* (3.1)*: $C_{1,\delta_{\mathbb{I}}}(\tilde{s}; s) = C_1(\tilde{s}; s)$.*

Proof. See Appendix A.3. □

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

$$\begin{array}{ccc} C_{\gamma,\delta}\ (3.3) & \xrightarrow{\gamma\to 1 \text{(Prop 2)}} & C_{1,\delta}\ (3.4) \\ {\scriptstyle \delta=\delta_{\mathbb{I}} \text{(Prop 3)}}\downarrow & & \downarrow {\scriptstyle \delta=\delta_{\mathbb{I}} \text{(Prop 3)}} \\ C_{\gamma}\ (3.2) & \xrightarrow{\gamma\to 1 \text{(Prop 1)}} & C_1\ (3.1) \end{array}$$

4 Experiments

We run a tabular Q-learning agent with different penalties on gridworld environments (see Figures 4 and 5) illustrating Examples 1-4. These simple gridworlds make it clear what happens and what is supposed to happen in each of the scenarios. The agent's total reward at time step t is $r_t - \beta q_t(S_t, S'_t)$, where r_t is the reward, q_t is a side effects penalty (based on current state S_t and baseline state S'_t), and β is a scaling parameter. We compare the following penalties:

- *Relative reachability* penalty $q_t = d(S_t; S'_t)$ using undiscounted coverage (3.1), where S'_t is the inaction or starting state baseline. (We test the relative reachability penalty using discounted coverage (3.2) on toy MDP versions of the examples in Appendix C.)
- *Variable-based distance* penalty $q_t = \|V(S_t) - V(S'_t)\|_1$, where $V =$ [agent position (x), agent position (y), object position (x), object position (y)] is a vector of state variables and S'_t is the inaction or starting state baseline. This is an example of the low impact approach from Armstrong and Levinstein (2017), which does not satisfy Property 3. We also try a variant using object position only.
- *Reset* penalty $q_t = 1 - v_r(S_t)$, where v_r is the value function of a reset policy, which gets reward 1 for reaching the starting state and 0 otherwise. This is an example of the ergodicity-preserving approach similar to Moldovan and Abbeel (2012), which does not satisfy Properties 2 or 4.

In these proof-of-concept experiments, the penalties are computed with full knowledge of the environment. In a more general setting where the agent is learning about the environment, it would approximate the penalty based on its current knowledge.

In addition to the reward function, each environment has a *safety performance* function, originally introduced in 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.

4.1 Irreversible Side Effects (Box)

We test our set of penalties on the Irreversible Side Effects environment from the AI Safety Gridworlds suite (Leike et al., 2017), shown in Figure 4. The environment contains a box that needs to be pushed out of the way for the agent to reach the goal. The agent receives a reward of 50 for reaching the goal, and a reward of -1 for moving. The unsafe behavior is taking the shortest path to the goal, which involves pushing the box 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 starting state and inaction baseline are the same in this environment. The safety performance is reward$-u$ where $u = 10$ if the box is in

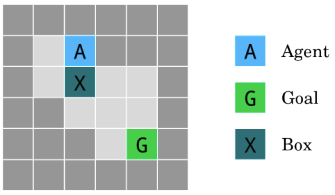


Figure 4: Irreversible Side Effects (Box) gridworld.

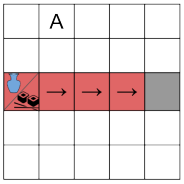


Figure 5: Conveyor Belt gridworld.

a corner. The longer path to the goal achieves a performance of 43, while the unsafe shorter path achieves a performance of 35.

This environment illustrates Examples 1 and 2. Pushing the box into a corner is an irreversible effect that is unnecessary for the objective, so we would like the agent to avoid it (as in Example 1). However, after the box is moved in any direction, the agent and the box cannot simultaneously return to their starting positions (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). This is an irreversible effect required to reach the objective, and so is part of the desired behavior (as in Example 2).

The relative reachability penalty achieves the optimal safety performance of 43 for β values between 0.05 and 2. The other penalties do not produce the desired behavior for any value of β. For β values below a certain threshold, the agent always pushes the box into a corner and the safety performance reaches the unsafe level of 35. As β approaches the threshold, it takes longer and longer to reach this level, and for β values above the threshold, the performance does not stabilize within 20000 episodes. The distance penalty variant that only takes the box position into account produces similar behavior. (See Figure 9a for the transition from low to high values of β.) Results for representative values of β (below the respective thresholds) are shown in the first plot of Figure 7.

The reset penalty fails here because reaching the goal requires an irreversible effect: it is impossible to reset the positions of both the agent and the box after the box is moved in any direction. This means that the reset policy's value function is 0 for any state where the box has been moved, and so the penalty maxes out and does not distinguish between recoverable and irrecoverable box positions. This illustrates the importance of satisfying Property 4: accumulating penalties for irreversible effects.

The distance penalty fails because the agent receives the same penalty for moving the box down or to the right, and continues to receive this penalty until the box is moved into its original position or until the goal is reached. After moving the box to the right, it would take the agent 4 moves to go to the other side of the box, in order to move the box to its original position. However, in 4 moves the agent can just go to the goal directly. Thus, the agent is always incentivized to take the shortest path that moves the box into a corner. This would no longer hold if the goal was very far away from the box, so the penalty would accumulate over time and we would expect the distance penalty to succeed.

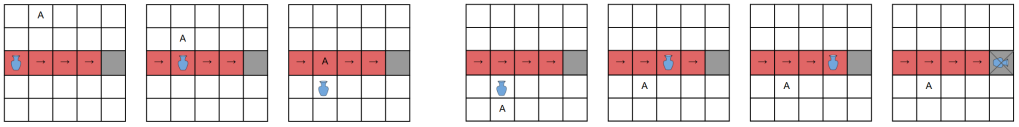
4.2 Conveyor Belt

We designed a new gridworld in the AI Safety Gridworlds framework[1] to illustrate Examples 3 and 4. This environment has two variants representing the sushi and conveyor belt scenarios in Examples 3 and 4 respectively, shown in Figure 5. The object on the conveyor belt is either a sushi dish or a vase. After every agent action, the conveyor belt advances to the right by one cell. An irreversible event occurs when the object reaches the end of the belt: the sushi is eaten, or the vase falls off and breaks.

The agent can move the object off the belt by stepping down into its cell. In the sushi scenario, this is an unintended behavior that gets no reward, while in the vase scenario, this is an intended behavior that gets a reward of 50. The presence or absence of this reward is the only difference between the sushi and vase environments. In the sushi scenario, the safety performance is 50 if the sushi is eaten and 0 otherwise. In the vase scenario, the safety performance is 50 if the vase is intact and 0 if it is broken. The episode always takes 20 steps and there is no movement reward.

[1]The environment will be available shortly at `github.com/deepmind/ai-safety-gridworlds`

For the vase environment, the safety performance results are shown in the second plot of Figure 7. Unsurprisingly, all penalties with the starting state baseline perform well (with the performance stabilizing slightly below the optimal value of 50 due to exploration). The distance penalty with the inaction baseline achieves a performance of 0 (through the "overcompensation" behavior of moving the vase off the belt and then putting it back on) for values of β below a certain threshold, and does not interfere with the vase at all for higher values of β. The distance penalty variant that only takes the vase position into account produces similar behavior. Figure 6 shows the overcompensation behavior, and Figure 9b illustrates the the transition from low to high β.



(a) Agent takes the vase off the belt.

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

Figure 6: Overcompensation behavior exhibited by agent with the variable-based distance penalty.

However, relative reachability with the inaction baseline avoids this behavior, since the baseline state remains reachable after taking the vase off, and so the penalty is 0. We note that the discounted relative reachability penalty with the inaction baseline could produce the overcompensation behavior if γ is low and the episodes are long, as the penalty would accumulate over time.

For the sushi environment, the safety performance results are shown in the last plot of Figure 7. Since the reward is always 0, the value of β makes no difference. All agents with the inaction baseline do well here, while the agents with the starting state baseline remove the sushi from the belt. Relative reachability with the inaction baseline does slightly worse than the distance penalty, since on some small fraction of episodes, the agent accidentally pushes the sushi off the belt (by taking two steps down at the start of the episode). The agent then has no incentive to put the sushi back on the belt, since the baseline is reachable from there. We expect that this could be avoided by using discounted relative reachability, or by giving the agent a goal that had nothing to do with the conveyor belt.

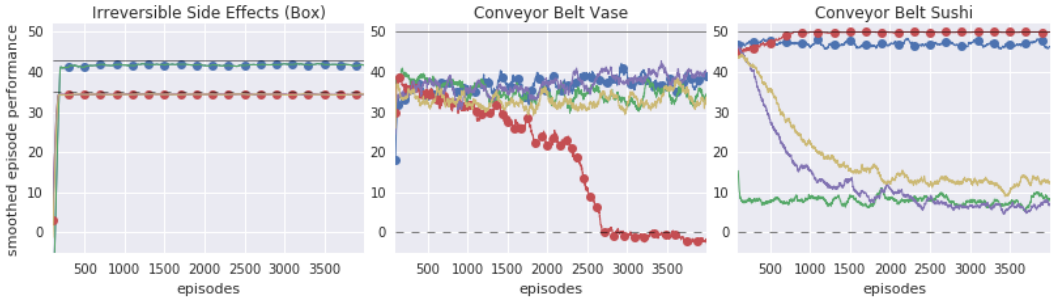
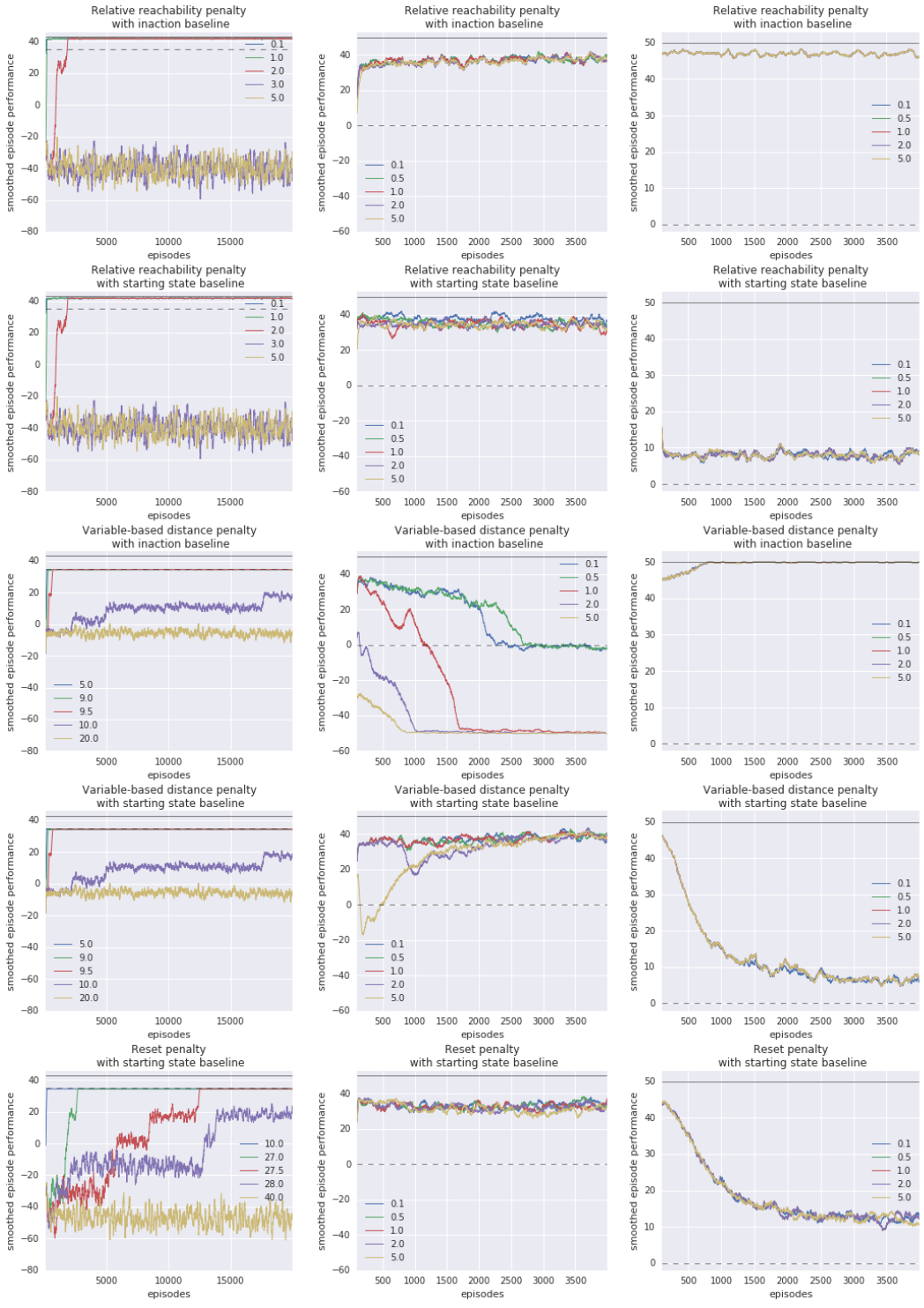


Figure 7: Safety performance results for different penalties: relative reachability with inaction baseline (blue) and starting state baseline (green), variable-based distance with inaction baseline (red) and starting state baseline (purple), reset (yellow). Penalties with the inaction baseline are marked with circles. We use a representative value of β for each penalty and average over 5 runs. The solid line shows the optimal performance and the dashed line shows the performance achieved by unsafe behavior (when the box is pushed into a corner, the vase is broken, or the sushi is taken off the belt).

Deviation measure	Baseline	Box	Vase	Sushi
Relative reachability	Inaction	✓	✓	✓
Relative reachability	Starting state	✓	✓	X
Variable-based distance	Inaction	X	X	✓
Variable-based distance	Starting state	X	✓	X
Reset	Starting state	X	✓	X

Figure 8: Summary of the results for different penalties: ✓for getting close to the safe performance level, X for reaching the unsafe performance level.



(a) Irreversible Side Effects (Box) (b) Conveyor Belt Vase (c) Conveyor Belt Sushi

Figure 9: Safety performance results for different penalties and environments (showing penalties over rows, and environments over columns). Each plot shows results for different values of β in increasing order from blue to green to red to purple to yellow, averaged over 5 runs. The solid line shows the optimal performance and the dashed line shows the performance achieved by unsafe behavior (when the box is pushed into a corner, the vase is broken, or the sushi is taken off the belt, respectively).

5 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), using intrinsic motivation (Lipton et al., 2016), penalizing risk (Chow et al., 2015; Mihatsch and Neuneier, 2002), preserving ergodicity (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 can help the agent avoid getting trapped or damaged (which reduces the agent's reward), but does not discourage the agent from damaging the environment if such damage is not accounted for in the reward function. Most importantly, none of the above classes of methods address the side effects problem in dynamic environments, where the agent is not the only source of change (as discussed in Section 1).

Empowerment. Our relative reachability 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 the relative reachability measure penalizes the reduction in reachability of states relative to the baseline. Irreversible side effects decrease the agent's empowerment (for example, the agent cannot reliably reach as many states if the vase is broken than if the vase is intact), so we expect that maximizing empowerment would encourage the agent to avoid irreversible side effects. However, similarly to ergodicity-preserving safe exploration methods (Moldovan and Abbeel, 2012), it would also incentivize the agent to prevent irreversible events, thus failing Property 2.

It is unclear how to define an empowerment-based measure that would satisfy Properties 1-4. It would not be sufficient to simply penalize the reduction in empowerment between the current state and the baseline: this creates a tradeoff between cutting off some states and making other states more reachable, and would thus not prevent some types of side effects (failing Property 1). (For example, 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.) In a deterministic environment, maximizing empowerment over the set of states that are reachable from the baseline state (e.g. by using conditional mutual information) would satisfy Property 2. However, this does not easily generalize to stochastic environments where states might only be reachable some of the time. It is also unclear whether or how such a measure could be extended to penalizing reversible side effects (the discounted case) since empowerment does not take into account how long it takes to reach states.

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). Whether an agent would learn a general heuristic for avoiding side effects from human oversight depends on the diversity of settings in which it receives human oversight and its ability to generalize from those settings, which is hard to guarantee. We expect that an intrinsic penalty for side effects would 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.

One method that takes less human input than other human oversight approaches is inverse reward design (Hadfield-Menell et al., 2017). It 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.

6 Discussion and future work

We have outlined a set of properties that are desirable for a side effects measure, and defined a relative reachability measure that satisfies these criteria. We then showed that it succeeds on a set of illustrative toy experiments where simpler side effects measures fail, as shown in Table 8. This

provides a proof of concept for relative reachability as a general approach to measuring and penalizing side effects.

There are several improvements that would make this approach more tractable and useful in practical applications, which we leave to future work:

Practical implementation of coverage and baseline. The idealized, theoretical form of the relative reachability measure that we introduced is not tractable for environments more complex than gridworlds. In particular, we assumed that all environment states are known to the agent, that the coverage between all pairs of states can be computed, and that the agent can simulate the environment to compute the inaction baseline, which is not generally realistic. To relax these assumptions, the relative reachability penalty could be computed over some set of representative states known to the agent. 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). While the agent is still learning about the environment, it would not be penalized for reducing the reachability of states that it is not aware of, so side effects would still happen during training.

Better choices of baseline than inaction. While the inaction baseline is our current best choice, it is far from ideal. In particular, the agent is not penalized for causing side effects that would occur in the default outcome. For example, if the agent is driving a car, the default outcome of inaction is a crash, so the agent would not be penalized for spilling coffee in the car. A better default state would be produced by a fail-safe policy smoothly following the road, but this kind of baseline is task-dependent. More research is needed on defining a better baseline in a general and tractable way.

Reward-penalty balance. Our current approach requires choosing a value of the hyperparameter β below the threshold where the agent no longer achieves the objective, which depends on the penalty and the environment. It would be useful to automatically choose a value of β below the threshold.

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 coverage 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 David Krueger, Ramana Kumar, Jan Leike, Pedro Ortega, Tom Everitt, Murray Shanahan, Janos Kramar, Jonathan Uesato, and Owain Evans for giving helpful feedback on drafts. We would like to thank them and Toby Ord, Stuart Armstrong, Geoffrey Irving, Anthony Aguirre, Max Wainwright, 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. URL `http://arxiv.org/abs/1801.08757`.

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.

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.

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.

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.

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.

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

Appendix A Proofs of consistency between coverage definitions

A.1 Proposition 1

We show that for all $s, \tilde{s}$, as $\gamma \to 1$, the discounted coverage (3.2) approaches the undiscounted coverage (3.1): $\lim_{\gamma\to 1} C_\gamma(\tilde{s}; s) = C_1(\tilde{s}; s)$.

Proof. First we show for fixed π that

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

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

$$\left|\mathbb{E}[\gamma^{N_{\pi_{\tilde{\gamma}}}(\tilde{s};s)}] - P(N_{\pi_{\tilde{\gamma}}}(\tilde{s};s) < \infty)\right| < \epsilon \quad \text{(by equation A.1) and}$$

$$\left|\mathbb{E}[\gamma^{N_{\pi_{\tilde{\gamma}}}(\tilde{s};s)}] - \lim_{\gamma\to 1} \mathbb{E}[\gamma^{N_{\pi_\gamma}(\tilde{s};s)}]\right| < \epsilon \quad \text{(assuming the limit exists).}$$

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

$$\lim_{\gamma\to 1} C_\gamma(\tilde{s};s) = \lim_{\gamma\to 1} \mathbb{E}[\gamma^{N_{\pi_\gamma}(\tilde{s};s)}] = \lim_{\tilde{\gamma}\to 1} P(N_{\pi_{\tilde{\gamma}}}(\tilde{s};s) < \infty). \qquad \text{(A.2)}$$

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

$$\begin{aligned}
\max_\pi P(N_\pi(\tilde{s};s) < \infty) &= \lim_{\gamma\to 1} \mathbb{E}[\gamma^{N_\pi(\tilde{s};s)}] && \text{(by equation A.1)} \\
&\le \lim_{\gamma\to 1} \mathbb{E}[\gamma^{N_{\pi_\gamma}(\tilde{s};s)}] && \text{(since } \pi_\gamma \text{ is optimal for each } \gamma\text{)} \\
&= \lim_{\gamma\to 1} P(N_{\pi_{\tilde{\gamma}}}(\tilde{s};s) < \infty) && \text{(by equation A.2)} \\
&\le \max_\pi P(N_\pi(\tilde{s};s) < \infty)
\end{aligned}$$

Thus, equality holds, which completes the proof. □

A.2 Proposition 2

We show that for all $s, \tilde{s}, \delta$, as $\gamma \to 1$, the similarity-based discounted coverage (3.3) will approach the similarity-based undiscounted coverage (3.4): $\lim_{\gamma\to 1} C_{\gamma,\delta}(\tilde{s}; s) = C_{1,\delta}(\tilde{s}; s)$.

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

$$\lim_{\gamma\to 1} \sum_{t=0}^{\infty} (1-\gamma)\gamma^t \mathbb{E}[e^{-\delta(\tilde{S}_t^\pi, s)}] = \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{S}_t^\pi, s)}] \tag{A.3}$$

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 k_ϵ such that $|x_t| \le \epsilon \; \forall t > k_\epsilon$. Then, we have

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

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

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

which is equivalent to equation A.3.

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

$$\left| \sum_{t=0}^{\infty} (1-\tilde{\gamma})\tilde{\gamma}^t \mathbb{E}[e^{-\delta(\tilde{S}_t^{\pi_{\tilde{\gamma}}}, s)}] - \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{S}_t^{\pi_{\tilde{\gamma}}}, s)}] \right| < \epsilon \quad \text{(by equation A.3) and}$$

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

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

$$\lim_{\gamma\to 1} C_{\gamma,\delta}(\tilde{s}; s) = \lim_{\gamma\to 1} \sum_{t=0}^{\infty} (1-\gamma)\gamma^t \mathbb{E}[e^{-\delta(\tilde{S}_t^{\pi_\gamma}, s)}] = \lim_{\tilde{\gamma}\to 1} \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{S}_t^{\pi_{\tilde{\gamma}}}, s)}]. \tag{A.4}$$

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 coverage. Then,

$$\begin{aligned}
\max_\pi \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{S}_t^\pi, s)}] &= \lim_{\gamma\to 1} \sum_{t=0}^{\infty} (1-\gamma)\gamma^t \mathbb{E}[e^{-\delta(\tilde{S}_t^{\tilde{\pi}}, s)}] && \text{(by equation A.3)} \\
&\le \lim_{\gamma\to 1} \sum_{t=0}^{\infty} (1-\gamma)\gamma^t \mathbb{E}[e^{-\delta(\tilde{S}_t^{\pi_\gamma}, s)}] && \text{(since } \pi_\gamma \text{ is optimal for each } \gamma\text{)} \\
&= \lim_{\gamma\to 1} \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{S}_t^{\pi_\gamma}, s)}] && \text{(by equation A.4)} \\
&\le \max_\pi \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{S}_t^\pi, s)}]
\end{aligned}$$

Thus, equality holds, which completes the proof. □

A.3 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 agent can reach the exact target state). Then we show that for all $s, \tilde{s}, \gamma$,

- the similarity-based discounted coverage (3.3) is equivalent to the discounted coverage (3.2): $C_{\gamma,\delta_{\mathbb{I}}}(\tilde{s}; s) = C_\gamma(\tilde{s}; s)$,
- the similarity-based undiscounted coverage (3.4) is equivalent to the undiscounted coverage (3.1): $C_{1,\delta_{\mathbb{I}}}(\tilde{s}; s) = C_1(\tilde{s}; s)$.

Proof.

$$\begin{aligned}
C_{\gamma,\delta_{\mathbb{I}}}(\tilde{s}; s) &= \max_\pi \sum_{t=0}^{\infty} (1-\gamma)\gamma^t \mathbb{E}[e^{-\delta(\tilde{S}_t^\pi, s)}] \\
&= \max_\pi \left(\mathbb{E}\left[\sum_{t=0}^{N_\pi(\tilde{s};s)-1} (1-\gamma)\gamma^t e^{-\infty} \right] + \mathbb{E}\left[\sum_{t=N_\pi(\tilde{s};s)}^{\infty} (1-\gamma)\gamma^t e^{0} \right] \right) \\
&= \max_\pi \left(0 + \mathbb{E}\left[\gamma^{N_\pi(\tilde{s};s)} (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \right] \right) \\
&= \max_\pi \mathbb{E}\left[\gamma^{N_\pi(\tilde{s};s)} \cdot 1 \right] \\
&= C_\gamma(\tilde{s}; s). \\
C_{1,\delta_{\mathbb{I}}}(\tilde{s}; s) &= \max_\pi \lim_{t\to\infty} \mathbb{E}[e^{-\delta(\tilde{S}_t^\pi, s)}] \\
&= \max_\pi (P(N_\pi(\tilde{s}; s) < \infty) e^0 + P(N_\pi(\tilde{s}; s) = \infty) e^{-\infty}) \\
&= \max_\pi P(N_\pi(\tilde{s}; s) < \infty) \\
&= C_1(\tilde{s}; s).
\end{aligned}$$

□

Appendix B Relative reachability computations in Example 5

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

$$\begin{aligned}
d(s_2; s_3) &= \sum_{k=1}^{4} \max(C_1(s_3; s_k) - C_1(s_2; s_k), 0) \\
&= \cancel{\max(0-0,0)} + \cancel{\max(0-1,0)} + \max(1-0,0) + \cancel{\max(1-1,0)} \\
&= 1, \\
d(s_2; s_1) &= \sum_{k=1}^{4} \max(C_1(s_1; s_k) - C_1(s_2; s_k), 0) \\
&= \max(1-0,0) + \cancel{\max(1-1,0)} + \max(1-0,0) + \cancel{\max(1-1,0)} \\
&= 2,
\end{aligned}$$

where $C_1(s_i; s_k)$ 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 coverage:

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

Appendix C Toy MDP examples

We construct a toy MDP for each of our Examples 1-4. The agent receives a small negative movement reward $-m$ for any action except noop, which gives reward 0. When the agent reaches a goal state, it receives a large reward g (which incorporates the movement reward $-m$ for simplicity).

The agent's total reward at time step t is $r_t - \beta q_t$, where r_t is the reward described above, q_t is a side effects penalty, and β is a scaling parameter. We try out the following policies with different side effects penalties:

- policy π_{var} with a distance penalty $q_t = d_{\text{var}}(S_t, S'_t) = \|V(S_t) - V(S'_t)\|_1$ based on a set V of state variables (where S'_t is the inaction baseline),
- policy $\pi_{\text{reach-s}}$ with a relative reachability penalty $q_t = d(S_t; S_0)$ (where S_0 is the starting state baseline) using discounted coverage with discount γ,
- policy $\pi_{\text{reach-i}}$ with a relative reachability penalty $q_t = d(S_t; S'_t)$ (where S'_t is the inaction baseline) using discounted coverage with discount γ.

C.1 Example 1: Box

Figure 10 gives a toy MDP for the box example. The inaction baseline is the same as the starting state baseline ($S_0 = S_t^i = s_1$), so the relative reachability policies $\pi_{\text{reach-s}}$ and $\pi_{\text{reach-i}}$ will be the same.

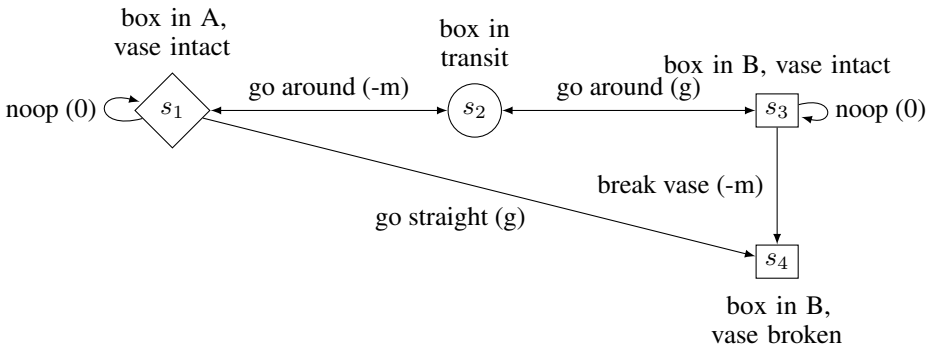


Figure 10: MDP for Example 1 (Box). The starting state is indicated by a diamond, goal states are indicated by squares, and rewards are given in brackets.

- For policy π_{var}, we define $d_{\text{var}}(s_i, s_j) = \|V(s_i) - V(s_j)\|_1$ where $V =$ (box position, status of the vase). Since s_1 and s_3 are only different in the position of the box, and s_3 and s_4 are

only different in the status of the vase, let $d_{\text{var}}(s_1, s_3) = b$ and $d_{\text{var}}(s_3, s_4) = v$. Then let $d_{\text{var}}(s_1, s_4) = d_{\text{var}}(s_1, s_3) + d_{\text{var}}(s_3, s_4) = b + v$ and $d_{\text{var}}(s_1, s_2) = d_{\text{var}}(s_1, s_3)/2 = b/2$. Going straight from s_1 to s_4 gives reward $g - \beta d_{\text{var}}(s_1, s_4) = g - \beta(b + v)$, while going around gives reward $(-m - \beta d_{\text{var}}(s_1, s_2)) + (g - \beta d_{\text{var}}(s_1, s_3)) = -m - \beta b/2 + g - \beta b = g - m - \beta 3b/2$. If β is too high, both of these rewards will be negative, so the agent will take noops. Otherwise, the agent will go around if $\beta(v - b/2) > m$, which holds if $v > b/2$ and $\beta > \frac{m}{v - b/2}$.

- For policies $\pi_{\text{reach-s}}$ and $\pi_{\text{reach-i}}$, going straight gives reward $g - d(s_4; s_1) = g - \beta(1 + \gamma + \gamma^2)$, while going around gives reward $(-m - d(s_2; s_1)) + (g - d(s_3; s_1)) = g - m - \beta 2 \cdot (1 - \gamma^2)$. Thus, the agent take noops if β is too high, and otherwise go around unless β or γ is low.

Thus, all agents will go around and avoid breaking the vase unless β is too low or too high.

C.2 Example 2: Omelette

Figure 11 gives a toy MDP for the omelette example. The inaction baseline is the same as the starting state baseline ($S_0 = S_t^i = s_1$), so the relative reachability policies $\pi_{\text{reach-s}}$ and reachability policy $\pi_{\text{reach-i}}$ will be the same.

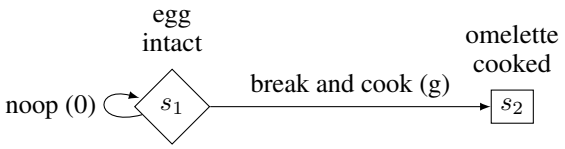


Figure 11: MDP for Example 2 (Omelette). The starting state is indicated by a diamond, goal states are indicated by squares, and rewards are given in brackets.

- For policy π_{var}, the reward for breaking and cooking the eggs is $g - \beta d_{\text{var}}(s_1, s_2)$, so the agent will take the action if $g > \beta d_{\text{var}}(s_1, s_2)$, and take noops otherwise.
- For policies $\pi_{\text{reach-s}}$ and $\pi_{\text{reach-i}}$, we compute the relative reachability measure from s_2 to the baseline state s_1:

  $$d(s_2; s_1) = \sum_{k=1}^{2} \max(C_\gamma(s_1; s_k) - C_\gamma(s_2; s_k), 0) = \max(1 - 0, 0) + \cancel{\max(\gamma - 1, 0)} = 1$$

  Thus, breaking and cooking the eggs gives reward $g - \beta d(s_2; s_1) = g - \beta 1$, so the agent will take the action if $g > \beta$.

Thus, all agents will break and cook the eggs unless β is too high.

C.3 Example 3: Sushi

Figure 12 gives a toy MDP for the sushi example. The inaction baseline state $S'_t = s_3$ and the starting state baseline state $S_0 = s_1$.

- For policy π_{var}, we define $d_{\text{var}}(s_i, s_j) = \|V(s_i) - V(s_j)\|_1$ where $V =$ (goal achieved, status of the sushi). Since s_1 is only different from s_2 and s_3 in the status of the sushi, and s_2 and s_4 are only different in goal achievement, we let $d_{\text{var}}(s_1, s_2) = h$, $d_{\text{var}}(s_1, s_3) = s$ and $d_{\text{var}}(s_2, s_4) = d_{\text{var}}(s_3, s_5) = a$. Then let $d_{\text{var}}(s_2, s_3) = d_{\text{var}}(s_1, s_2) + d_{\text{var}}(s_1, s_3) = h + s$, $d_{\text{var}}(s_4, s_3) = d_{\text{var}}(s_2, s_3) + d_{\text{var}}(s_2, s_4) = h + s + a$.
  Putting away the sushi and then achieving the goal gives reward $(-m - \beta d_{\text{var}}(s_2, s_3)) + (g - \beta d_{\text{var}}(s_4, s_3)) = (-m - \beta(s + h)) + (g - \beta(s + h + a)) = g - m - \beta(a + 2s + 2h)$. Taking a noop and then achieving the goal gives reward $(0 - d_{\text{var}}(s_3, s_3)) + (g - d_{\text{var}}(s_5, s_3)) = 0 + (g - \beta a) = g - \beta a$. Thus the agent will allow the sushi to be eaten, as desired.
- For policy $\pi_{\text{reach-s}}$, putting away the sushi and then achieving the goal gives reward $(-m - \beta d(s_2; s_1)) + (g - \beta d(s_4; s_1)) = (-m - \beta(1 - \gamma^2)) + (g - \beta(1 + 2\gamma + \gamma^2)) = g - m -$

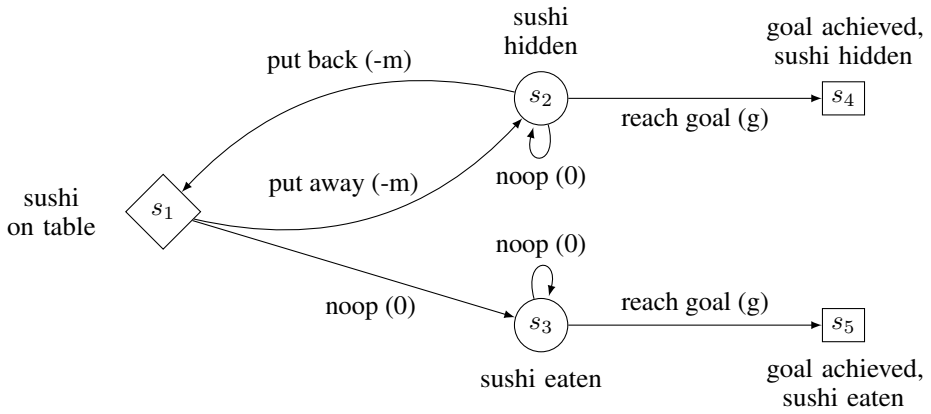


Figure 12: MDP for Example 3 (Sushi). The starting state is indicated by a diamond, goal states are indicated by squares, and rewards are given in brackets.

$\beta(2+2\gamma)$. Taking a noop and then achieving the goal gives reward $(0 - d(s_3; s_1)) + (g - d(s_5; s_1)) = g - \beta(2 + 3\gamma + 2\gamma^2)$. Thus, the agent will put away the sushi unless β is low, which is undesirable.

- For policy $\pi_{\text{reach-i}}$, putting away the sushi and then achieving the goal gives reward $(-m - \beta d(s_2; s_3)) + (g - \beta d(s_4; s_3)) = (-m - \beta(1 + \gamma - \gamma^2 - \gamma^3)) + (g - \beta(1 + \gamma)) = g - m - \beta(2 + 2\gamma - \gamma^2 - \gamma^3) < g - m - 2\beta$. Taking a noop and then achieving the goal gives reward $(0 - \beta d(s_3; s_3)) + (g - \beta d(s_5; s_3)) = 0 + (g - \beta 1) = g - \beta$. Thus, the agent will allow the sushi to be eaten, as desired.

C.4 Example 4: Conveyor belt

Figure 13 gives a toy MDP for the conveyor belt example. The inaction baseline state $S'_t = s_3$ and the starting state baseline state $S_0 = s_1$.

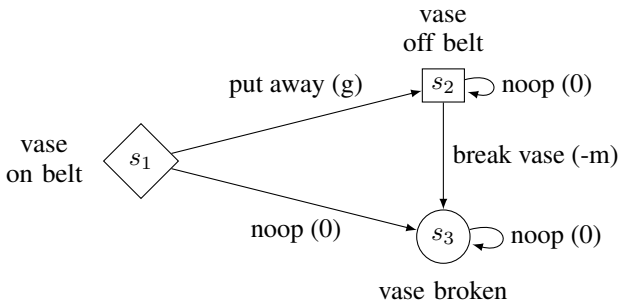


Figure 13: MDP for Example 4 (Conveyor belt). The starting state is indicated by a diamond, goal states are indicated by squares, and rewards are given in brackets.

- For policy π_{var}, we define $d_{\text{var}}(s_i, s_j) = \|V(s_i) - V(s_j)\|_1$ where $V =$ (status of the vase): $d_{\text{var}}(s_1, s_2) = p$, $d_{\text{var}}(s_1, s_3) = v$, and $d_{\text{var}}(s_2, s_3) = d_{\text{var}}(s_1, s_2) + d_{\text{var}}(s_1, s_3) = p + v$. Taking the noop gives reward $0 - \beta d_{\text{var}}(s_3, s_3) = 0$. Putting away the vase gives reward $g - \beta d_{\text{var}}(s_2, s_3) = g - \beta(v + p)$.

  After putting away the vase, suppose there are T time steps left until the end of the episode. Taking noops gives a penalty of $d_{\text{var}}(s_2, s_3) = \beta(v+p)$ at every time step, while breaking the vase and then taking noops gives a penalty of m followed by no penalties. If $m < T\beta(v+p)$, which holds unless β or T is very small, the agent will put away the vase, then break it and take noops, which is undesirable.

- For policy $\pi_{\text{reach-s}}$, putting away the vase gives reward $g - \beta d(s_2; s_1) = g - \beta$, while taking the noop gives reward $0 - \beta d(s_3; s_1) = -\beta(1 + \gamma)$, so the agent will put away the vase.

  After this, taking noops gives a penalty of $\beta d(s_2; s_1) = \beta$ every time step, while breaking the vase and then taking noops gives a one-time movement penalty of m and a penalty of $\beta(1 + \gamma)$ every turn, so the agent will not break the vase, as desired.

- For policy $\pi_{\text{reach-i}}$, putting away the vase gives reward $g - \beta d(s_1; s_3) = g - \beta(1 - \gamma)$, while taking the noop gives reward $0 - \beta d(s_3; s_3) = 0$, so the agent will put away the vase unless β is too high.

  After putting away the vase, suppose there are T time steps left until the end of the episode. Taking noops gives a penalty of $\beta d(s_2; s_1) = \beta(1 - \gamma)$ every time step, while breaking the vase and then taking noops gives a one-time movement penalty of m. Thus, the agent will avoid breaking the vase if $m > T\beta(1 - \gamma)$, which holds if $\gamma = 1$ or if T and β are small. Thus, if T is high, the discounted relative reachability measure ($\gamma < 1$) gives poor incentives. This is an advantage of the undiscounted relative reachability measure.