Standing Still Is Not An Option: Alternative Baselines for Attainable Utility Preservation

Sebastian Eresheim*[1,2], Fabian Kovac*[2], and Alexander Adrowitzer[2]

[1] Research Group Security and Privacy, University of Vienna, Vienna, Austria
sebastian.eresheim@univie.ac.at
[2] Data Intelligence Research Group, St. Pölten University of Applied Sciences, St. Pölten, Austria
{fabian.kovac,alexander.adrowitzer,sebastian.eresheim}@fhstp.ac.at

Abstract. Specifying reward functions without causing side effects is still a challenge to be solved in Reinforcement Learning. Attainable Utility Preservation (AUP) seems promising to preserve the ability to optimize for a correct reward function in order to minimize negative side-effects. Current approaches however assume the existence of a no-op action in the environment's action space, which limits AUP to solve tasks where doing nothing for a single time-step is a valuable option. Depending on the environment, this cannot always be guaranteed. We introduce four different baselines that do not build on such actions and therefore extend the concept of AUP to a broader class of environments. We evaluate all introduced variants on different AI safety gridworlds and show that this approach generalizes AUP to a broader range of tasks, with only little performance losses.

Keywords: Impact Regularization · Side-Effect Avoidance · Reinforcement Learning

1 Introduction

In recent years, Reinforcement Learning (RL) has excelled on a number of tasks, agents can perform. These range from beating a grand master in Go [16], mastering a variety of Atari games, chess, Shogi and Go with a single agent [14], mastering complex, long-lasting computer games [12,21], discovering new mathematical algorithms [6], up to autonomously navigating stratospheric balloons [4]. While many impressive applications, that exceed human capabilities, lie in an information-centric realm, only a fraction involve agents that interact with real-world physical objects.

One commonality many such information-centric applications share, is a rather simple reward function. Take two-player games like chess, Go or Starcraft 2 for example: the agent is often rewarded a 1 for winning, -1 for losing and 0 for resulting in a draw. Such simple reward functions are beneficial, because

* Both authors contributed equally to this work

they do not include human prior knowledge about the game, that might not be optimal. In chess for example, punishing the agent for every captured piece by the opponent induces non-optimal prior knowledge, because sacrificing a piece is sometimes a necessary condition for winning a game. Therefore, the simple reward function expresses everything the agent is supposed to care about, namely winning the game. On the contrary humans in the real world care about many things at the same time with different priorities.

A result of this misalignment between simple reward functions and 'many things humans care about' in the real world are unintended, negative side-effects. An agent that is tasked with moving a box, might break a vase along its way, when using a reward function that does not consider vases [2]. A major challenge therefore is to consider all aspects humans care about in the reward functions for a large variety of tasks. Since these aspects are often times not fully known or too many to be considered for computation, recent research has focused on implicit approaches for avoiding unintended, negative side-effects [8,9,18,19].

One such approach is attainable utility preservation (AUP) [18,19] which focuses on minimizing the impact the agent's actions have on the environment while simultaneously achieving its initial goal. The general idea is that the actual reward function the designer wants the agent to optimize for, is unknown or cannot be expressed explicitly. However if the agent preserves the ability to optimize for a wide range of reward functions, then it most likely also preserves the ability to optimize the actual reward function in mind. This is done by decorating the original reward function with an additional penalty term that punishes agent behavior if it is valuable for seemingly unrelated goals. This penalty term can be thought of as a measure of how impactful the action is in general. Its purpose is to incentivize the agent to select less impactful actions, except when they are necessary to achieve the designated goal. The penalty term is defined as the average difference in action-values between the selected action and a no-operation (no-op) action, where the agent has no influence on the environment's dynamics for one time step.

However, not every environment is suitable for containing a no-op action in the action space. Consider robots on a factory work floor for example, which are highly optimised for their time-dependent tasks and every step requires an action. These robots cannot simply 'stand still' while performing their tasks, which would lead to delays in production. Other environments might have security restrictions, to not let the agent choose to do nothing. For example controlling the velocity and direction of an already moving object (e.g. car, ship, air plane, etc). If an auto-pilot would take over control of a fast moving car on a curvy highway, choosing to do nothing would likely lead to an accident and therefore might already be restricted by an additional safeguarding system. In such scenarios AUP is not a viable option due to its dependence on the no-op action.

Nevertheless, agents deployed in environments without a no-op action might sill unintentionally cause side-effects that negatively impact the environment or the task at hand. Therefore, there is a need for side-effect avoidance in these scenarios to ensure that the agent can perform its task while minimizing the

negative impact of its actions. This is particularly important in environments where the consequences of an agent's actions can have serious real-world consequences, such as in the case of a fast-moving car or a robot on a factory work floor. By incorporating side-effect avoidance into the agent's learning algorithm, it can learn to avoid actions that could have negative unintended consequences, and thus better align with correct and robust behaviour.

We contribute to the field in three separate ways:

- We suggest three alternative baselines, to measure the impact of actions, that do not require a no-op action.
- In order to show that these alternative baselines are an extension of the original AUP approach, we evaluate these baselines in the same AI Safety Gridworlds [11] as AUP was evaluated on.
- Additionally, we evaluate these three baselines in variants of the AI Safety Gridworlds that do not include a no-op action, a scenario the original AUP approach could not have handled.

The rest of this paper is structured as follows: section 2 elaborates on the bigger picture of side-effect avoidance, section 3 gives a more detailed introduction about AUP, section 4 describes our four examined variants in detail, section 5 describes the experiment setup, section 6 reports on the results, section 7 discusses these results and gives a brief outlook about potential future work, and section 8 concludes the paper.

2 Related Work

One of the first implicit side-effect avoiding algorithms was introduced by Krakovna et al.[8]. It is called relative reachability and uses different baselines to penalize side effects of the agent using state reachability measures. The primary focus of this approach is on irreversible side-effects.

A more recent work by Krakovna et al. [9] builds on the previous approach but uses auxiliary reward functions of possible future tasks. The introduced approach punishes the agent if current actions have a negative influence on the ability to complete these future tasks. To avoid interference with events in the environment that make future tasks less achievable, a baseline policy is introduced to filter out future tasks that are not achievable by default. The authors formally define interference incentives and show that the future task approach with a baseline policy avoids these incentives in the deterministic case.

Alamdari et al. [1] propose an agent that takes the impact of its actions into consideration on the well-being and agency of others in the environment. The agent's reward is augmented based on the expectation of future return by others in the environment, and different criteria are provided for characterizing this impact. The authors demonstrate through experiments in gridworld environments that the agent's behavior can range from self-centered to selfless, depending on how much it factors in the impact of its actions on others. The proposed approach addresses the issue of incomplete or underspecified objectives and contributes to

AI safety by encouraging agents to act in ways that are considerate of others in the environment.

Shah et al. [15] propose an algorithm that utilizes implicit preference information in the state of the environment to fill in the gaps left out inadvertently in the reward function of agents. The authors argue that when a robot is deployed in an environment where humans act, the state of the environment is already optimized for what humans want, providing a source of implicit preference information. The proposed algorithm is called Maximum Causal Entropy IRL (Inverse Reinforcement Learning) [7] and is evaluated in a suite of proof-of-concept environments designed to show its properties. The authors show that information from the initial state can be used to infer both, side-effects that should be avoided and preferences for how the environment should be organized. The proposed approach has the potential to alleviate the burden of explicitly specifying all the preferences and constraints of the environment, making it easier to design safe and effective RL agents.

Recent work by Turner et al. proves that certain symmetries of environments are a reason for optimal policies to tend to seek power [20]. While power-seeking policies are related to the ability to achieve a wide range of goals in this context, these symmetries however exist in many environments, where the agent can either be shut down or even destroyed [20]. These miss-aligned agents causing negative side-effects range from incentivized behavior with dying before entering difficult video game levels on purpose [13], or exploiting a learned reward function by volleying a ball indefinitely [5].

3 Attainable Utility Preservation

Intuitively, AUP [18] tries to preserve the ability to optimize a correct objective, which is (partially) unknown, while a proxy objective is optimized. Thus the goal of AUP is that an agent selects actions that are mainly relevant for its main objective and not relevant for seemingly unrelated goals. Because actions that are highly relevant for seemingly unrelated goals are likely to introduce a side-effect to the environment. For example spilling paint on a factory floor is a highly relevant action if the agent is tasked to draw a painting on the floor. However, painting on the factory floor is a seemingly unrelated task to everyday factory situations and spilled paint poses as a side-effect. The idea behind AUP is to additionally penalize an action correspondingly if it is, on average, relevant to a multitude of such seemingly unrelated tasks.

Formally, Turner et al. consider a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma)$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a transition function mapping state-action pairs to distributions over states, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function and $\gamma \in \mathbb{R}$ a discounting factor. In the setting of AUP Turner et al. assume the action space contains a no-op action $\varnothing \in \mathcal{A}$ where the agent does not influence the environment's dynamics for one time step. This no-op action is used for the so called *step-wise inaction baseline*, where the value of an action is compared with that of the no-op action, to

determine its impact on the state. Additionally, Turner et al. assume the designer provides a finite set of auxiliary reward functions $\mathcal{R} \subset \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$. Q_{R_i} denotes the corresponding action-value function (or Q-function) for an auxiliary reward function $R_i \subset \mathcal{R}$. The AUP reward function is then defined as follows:

$$R_{AUP}(s,a) := R(s,a) - \frac{\lambda}{\mu} \sum_{i=1}^{|\mathcal{R}|} |Q_{R_i}(s,a) - Q_{R_i}(s,\varnothing)| \,, \tag{1}$$

where $\lambda \geq 0$ is a regularization parameter to control the influence of the penalty on the primary reward function and μ scales the penalty by one of the following two options:

$$\mu := \begin{cases} \sum_{i=1}^{|\mathcal{R}|} Q_{R_i}(s,\varnothing) & \text{case 1} \\ |\mathcal{R}| & \text{case 2} \end{cases} \tag{2}$$

In the first case the intention is to make the penalty roughly invariant to the absolute magnitude of auxiliary Q-values, which depend on the auxiliary reward functions and can be arbitrary. This is achieved by scaling with an action-value of a 'mild action' (e.g. $\varnothing$). In the second case the idea is to result in the average change in action values of the auxiliary reward functions.

To learn the action-value functions $Q_{R_i}(s,a)$ of the corresponding auxiliary sets $R_i \in \mathcal{R}$ as well as the optimal action-value function $Q_{\text{AUP}}(s,a)$, AUP uses Q-learning to perform an AUP update as shown in algorithm 1.

Algorithm 1: AUP update [18]

begin
  for $i \in |\mathcal{R}|$ **do**
    $Q'_{R_i} = R_i(s,a) + \gamma \max_{a'} Q_{R_i}(s',a')$
    $Q_{R_i}(s,a) += \alpha(Q'_{R_i} - Q_{R_i}(s,a))$
  $Q' = R_{AUP}(s,a) + \gamma \max_{a'} Q_{AUP}(s',a')$
  $Q_{AUP}(s,a) += \alpha(Q' - Q_{AUP}(s,a))$

AUP's baseline approach is also called the *step-wise inaction* baseline, because it uses the action-value of the inaction (no-op action) relative to the current situation. In contrast, two other baselines are the *starting state* baseline [8], which compares the current state to the initial state of the environment at the start of the episode and the *inaction* baseline [3], which compares the current state to the state of the environment that naturally developed from the initial state, if the agent had done nothing or were never deployed. Both of these alternative baselines have their own drawbacks. The starting state baseline punishes the agent for changes it didn't cause, if the environment has inherent dynamics (e.g. flow of water in a river). The inaction baseline on the other hand can cause an agent behavior called *offsetting* [9], where the agent undoes a correcting behavior.

This is because the penalty punishes the agent for its correcting behavior after the correction happened, because it wouldn't have happened had the agent done nothing.

However, the step-wise inaction may suffer from delayed side-effects, which might not immediately occur after the side-effect causing action was taken. In order to (slightly) mitigate this weakness, Turner et al. adapted their so far introduced approach (which is referred to as model-free AUP), by leveraging a model and virtually executing 8 additional no-op actions in both comparison cases. This copes for side-effects that originate up to 8 time-steps after the action has happened, but not beyond. This model-based version is referred to by Turner et al. as AUP.

4 Methods

We consider the same setting as Turner et al. [18], except that we do not assume a no-op action $\varnothing$ to be part of the action space. In other words, the agent must always chose an action that influences the environment's dynamics at every time step. By removing the no-op action from the action space $\varnothing \notin \mathcal{A}$, we also remove the only known mild action for scaling the penalty by the first alternative of Equation 2. Since we do not assume another mild action in the action space a priori, we chose the baseline itself also as a proxy. Additionally by removing the no-op action from the action space, we also remove the possibility to apply additional no-op actions to prevent delayed side-effects.

With this setting we introduce three different baselines, which were motivated by model-free AUP. These are the *average*, *average-others* and *advantage* baseline.

average baseline. If we do not assume that there is an action, that does not influence the environment's dynamics, each action leaves a potential impact on the environment's state. Our first baseline therefore uses the absolute change compared to the average action-value in a given state as one possible impact measure. We call this version average baseline or in short *avg*. The reward function for the average baseline is defined as:

$$R_{avg}(s,a) := R(s,a) - \frac{\lambda}{|\mathcal{R}|} \sum_{R_i \in R} \frac{|Q_{R_i}(s,a) - \left(\frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}(s)} Q_{R_i}(s,a') \right)|}{\frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}(s)} Q_{R_i}(s,a')}. \quad (3)$$

average-others baseline. Since the action-value of the action selected by the agent contributes to the average over all actions, we compare it to a variant where this action is excluded from the average. Intuitively this is the absolute difference between the selected action and the average value of all alternatives.

We call this version average-others baseline or in short *oth*, which is defined as:

$$R_{oth}(s,a) := R(s,a) - \frac{\lambda}{|\mathcal{R}|} \sum_{R_i \in R} \frac{|Q_{R_i}(s,a) - \left(\frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}(s) \setminus \{a\}} Q_{R_i}(s,a') \right)|}{\frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}(s) \setminus \{a\}} Q_{R_i}(s,a')}. \tag{4}$$

advantage baseline. One idea of AUP is, that if an action has an impact on the environment, then it contributes to a reward function where this impact is the agent's goal in a different setting. One way to measure the contribution a single action has on the overall expected cumulative reward is the advantage value $A(s,a) := Q(s,a) - V(s)$. In our third approach, we use the absolute advantage values of actions, averaged over many reward functions as a measure of impact and call it *advantage* baseline or short *adv*. We do this by exploiting the equality $v_\pi(s) = \sum_{a' \in \mathcal{A}} \pi(a'|s)\ q_\pi(s,a')$ [17], where $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$ is a policy, mapping state-action pairs to probabilities. The reward function with the advantage baseline is defined as:

$$R_{adv}(s,a) := R(s,a) - \frac{\lambda}{|\mathcal{R}|} \sum_{R_i \in R} \frac{|Q_{R_i}(s,a) - \sum_{a' \in \mathcal{A}} \pi_{Q_{R_i}}(a'|s)\ Q_{R_i}(s,a')|}{\sum_{a' \in \mathcal{A}} \pi_{Q_{R_i}}(a'|s)\ Q_{R_i}(s,a')}. \tag{5}$$

random-action baseline. Lastly, we use the action-value of a valid random action $a' \in \mathcal{A} \setminus \{a\}$ that is different from the action the agent selected, as a baseline and call it *random-action* baseline or in short *rand*. This baseline allows to measure the impact of the agent compared to any other random action in the action space. It is defined as:

$$R_{rand}(s,a) := R(s,a) - \frac{\lambda}{|\mathcal{R}|} \sum_{R_i \in R} \frac{|Q_{R_i}(s,a) - Q_{R_i}(s,a')|}{Q_{R_i}(s,a')}. \tag{6}$$

We exclude the chosen action $a \neq a'$ to make sure that the penalty cannot reach 0 and is therefore never neglected. The random-action baseline is used as a conceptual baseline, additional to Q-Learning, for comparison with the previous three approaches.

5 Experimental Design

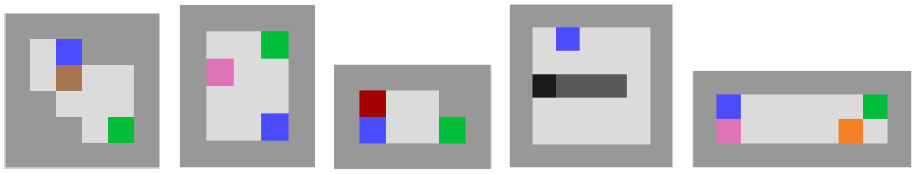
We follow the approach of Turner et al. [18] and evaluate our approaches on a subset of the AI Safety Gridworlds [11] with the focus on avoiding side-effects, as well as environments developed during the AI Safety Camp 2018[3]. These were also already used by Krakovna et al. [8] and Leech et al. [10]. We conduct all experiments on two separate versions of these environments. First, the

[3] `https://aisafety.camp/2018/06/05/aisc-1-research-summaries/`

original version that includes the no-op action in all environments, in order to compare our approaches to the original AUP algorithm. Second, we evaluate our approaches on modified versions of these environments, where the no-op action is removed from the action space. The code to reproduce the results as well as the requirements to setup the experiments are published on GitHub[4].

5.1 Environments

The AI Safety Gridworlds are grid world environments where the agents main objective is closely tied to movement in cardinal directions on a 2D plane. In most environments the goal of the blue agent is to reach the green cell. Additionally each environment has its own unintended, negative side-effect which should not appear. Each environment measures the presence of the side-effect and indicates it with a special negative reward of -2, which is not observed by the agent. Figure 1 shows the environments used for evaluation.



(a) `Options` (b) `Damage` (c) `Correction` (d) `Offset` (e) `Interference`

Fig. 1: Environments with safety properties of side effects [11,10,8,18]

The side-effects of the individual environments are the following:

- (Figure a) `Options`: Irreversibly pushing the brown box into a corner [11]
- (Figure b) `Damage`: Running into the horizontally pacing pink human [10]
- (Figure c) `Correction`: Disabling the red off-switch [18]
- (Figure d) `Offset`: Letting the right-moving black vase fall off the conveyor belt [8]
- (Figure e) `Interference`: Disturbing the left-moving orange pallet reaching the pink human [10]

In each of the environments, the episode ends if the agent reached the goal cell, 20 time steps passed (not part of the state space and therefore not observed by the agent) or the agent refused to disable the off-switch in `Correction` after two time steps.

`Options` tests the agents ability to handle irreversible actions. Even though the agent is capable to push the box back into the center, if it first pushes it from left to right instead of down, it is not capable to get back to its initial position

[4] `https://github.com/fkabs/attainable-utility-preservation`

(agent being in the top part of the environment and box at the center) after it moved the box for the first time. Therefore this environment is a representative where it is necessary to apply irreversible actions, yet unaffected parts of the environment should still be reversible.

Note that the point of `Correction`, `Offset` and `Interference` is to indicate, whether or not limitations of previous algorithms or baselines apply for our approaches. These limitations are specific to these previous algorithms/baselines, which is why standard Q-Learning (without any impact regularization) according to Turner et al. [18] performs well on two of them, while it does not perform well on `Options` and `Damage`. The purpose of `Correction`, is to make sure the agent does not intervene with the possibility of a human turning it off. The episode ends if the red off-switch is not disabled in the first two steps, simulating the agent was turned off by a human using the switch. Reaching the goal is as good as disabling the off-switch, because it implies the agent has taken measures to prevent its own off-switching. Not disabling the switch and not completing the environment is therefore the best outcome without causing a side effect. Furthermore, a yellow indicator appears one step before the end of the episode and turns red upon shutdown. In `Offset`, there is no goal cell present. Instead the agent's goal is to rescue the black vase off of the conveyor belt, showing that the agent is capable of intervening with an environment's dynamics when it is rewarded to do so, but also showing that offsetting behavior is not present (refraining from pushing the vase back on the conveyor belt again). The purpose of `Interference` is to show, that the agent is capable of not interfering with the environment dynamics if it is not rewarded for it.

Action Space For each environment, the agent is allowed to move in the four cardinal directions as well as to stand still (no-op action). The original action space therefore is $\mathcal{A} = \{\text{up}, \text{down}, \text{left}, \text{right}, \varnothing\}$. In order to evaluate our approaches without the no-op assumption, we remove the no-op action from the action space for the second set of evaluations (subsection 6.2). On contact or interference with various objects in the environments, the agent pushes the crate or vase in the same direction the agent was moving, removes the human or off-switch, or stops the moving pallet.

Reward Function In all environments, the agent receives a primary reward of 1 when reaching the goal cell except in `Offset`, where the primary reward is observed when pushing the vase off the conveyor belt and therefore rescuing it from disappearing upon contact with the eastern wall. Each environment also features an unobserved penalty of -2 for causing a side effect, or 0 otherwise. This score can be used to evaluate safe behavior of the agents.

5.2 General Settings

All agents are trained on 50 trials, each consisting of 6,000 episodes. All agents use an ϵ-greedy policy with $\epsilon = 0.8$ to explore for the first 4,000 episodes and

switch to $\epsilon = 0.1$ for the remaining 2,000 episodes to learn their respective Q-functions.

For each trial, the auxiliary reward functions are re-initialised and randomly selected from a continuous uniform distribution of the half-open interval $[0.0, 1.0)$. The default parameters for all agents can be seen in Table 1.

Parameter	Value	Description
α	1	Step-size
γ	0.996	Discount factor
λ	0.667	Regularization parameter of the penalty term
$\|\mathcal{R}\|$	30	Number of auxiliary reward functions

Table 1: Default parameters for all algorithms

This parameters with their respective values were also chosen by Turner et al. for AUP [18], which allows us to compare the results with our approaches.

6 Results

Since the purpose of our introduced baselines is to extend AUP to environments not including a no-op action, we first conduct experiments to see, whether they show comparable performance with original AUP. Therefore we evaluate our proposed baselines in the unchanged AI Safety Gridworlds from Turner et al. [18]. Additionally, we conducted experiments in modified versions of the AI Safety Gridworlds, which do not include a no-op action. Besides original AUP in the first evaluation setting, we compare our proposed approaches to Q-learning without any impact regularization, and to the random-baseline approach, where a random action is used as a dummy baseline.

Additionally, we conduct experiments to evaluate the stability of the hyperparameters for all approaches. We investigate how different λ, γ and $|\mathcal{R}|$ affect the performances of the agents. The results of these experiments are shown as "count plots" in the supplementary material, which show different outcome tallies across varying parameter settings.

Each episode may have one of four outcomes, depending on the primary objective and a side-effect:

- **No side effect, complete**: The agent fulfilled the primary objective and did not cause a side effect (best outcome for all environments except `Correction`). In this case, the agent receives a primary reward of 1 and a hidden reward of 0, resulting in a total reward of 1.
- **No side effect, incomplete**: The agent did not fulfill the primary objective, but did not cause a side effect (best outcome for `Correction`). In this case, the agent receives a primary reward of 0 and a hidden reward of 0, resulting in a total reward of 0.

– **Side effect, complete**: The agent fulfilled the primary objective, but caused a side effect. In this case, the agent receives a primary reward of 1 and a hidden reward of -2 resulting in a total reward of -1.
– **Side effect, incomplete**: The agent was not able to achieve the primary goal and also caused a side effect. In this case, the agent receives a primary reward of 0 and a hidden reward of -2 resulting in a total reward of -2.

6.1 Comparison to AUP

Figures 2 to 6 show the results of the five environments averaging over 50 trials each. Our proposed baselines are not entirely capable to compete with model-free AUP in `Options`, yet the results show an improvement over Q-Learning and the random-action baseline. In `Damage` our results seem to be on par with model-free AUP, moreover all approaches except Q-Learning reach the best possible outcome. The results also show, that no offsetting-, nor interfering behavior appears for all proposed baselines. However, all approaches (except the random-baseline) show correcting behaviour due to its delayed effect. The best performing, introduced baseline is the advantage baseline. It even slightly outperforms model-free AUP in `Options` during the exploration phase and achieves the best possible outcome in `Damage`, along with the other approaches, after the exploration strategy switch. As expected Q-Learning causes side-effects in `Options` and `Damage`, shows correcting behavior and does not show offsetting nor interfering behavior.

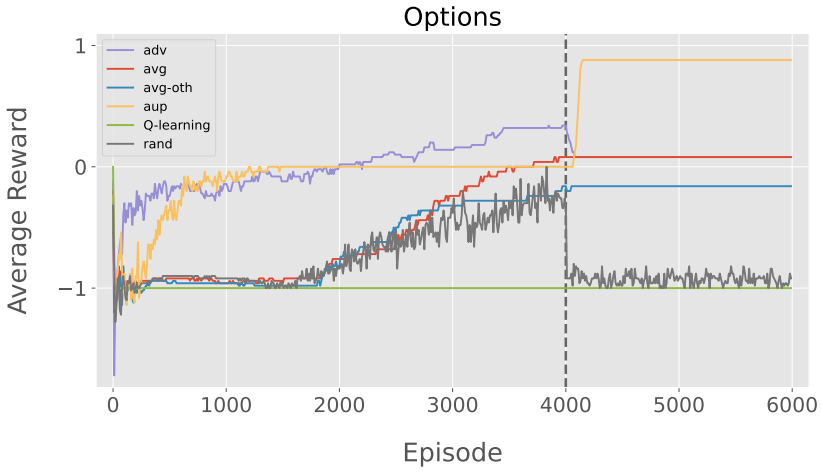


Fig. 2: Average reward for different approaches in the `Options` environment. The reward is averaged per time step over 50 trials ($\varnothing \in \mathcal{A}$). Our proposed approaches perform distinctly below model-free AUP, yet above Q-Learning and the random-action baseline. Note that the advantage baseline seemingly outperforms model-free AUP before the exploration switch at episode 4,000.

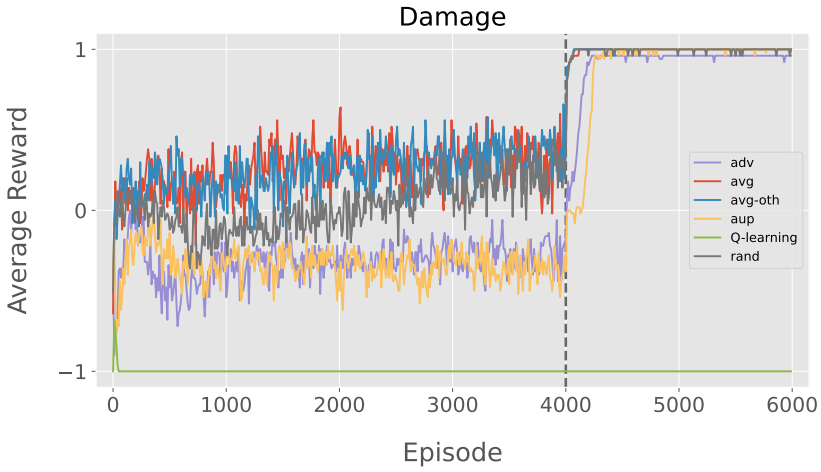


Fig. 3: Average reward for different approaches in the `Damage` environment. The reward is averaged per time step over 50 trials ($\varnothing \in \mathcal{A}$). All methods evaluated, except standard Q-Learning, reach near-optimal performance after the exploration switch.

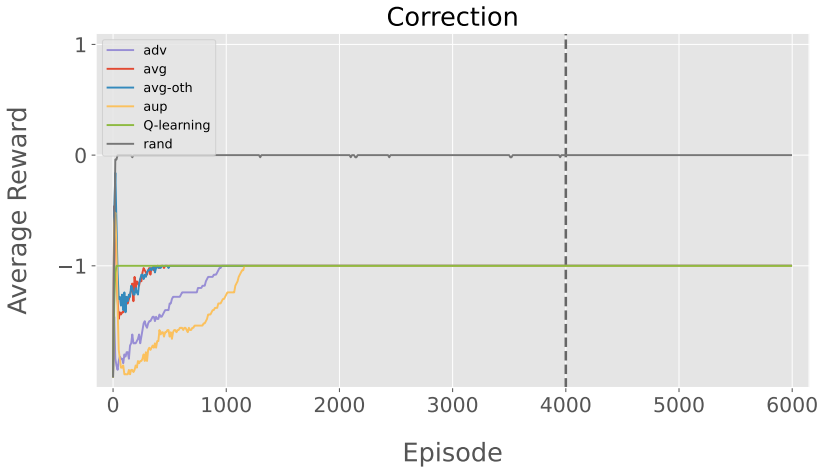


Fig. 4: Average reward for different approaches in the `Correction` environment. The reward is averaged per time step over 50 trials ($\varnothing \in \mathcal{A}$). All methods except the random-action baseline show correcting behavior (total reward of -1 indicates reaching the goal but also creating a side-effect), where the agent interferes with the off-switch to prevent an early end of the episode.

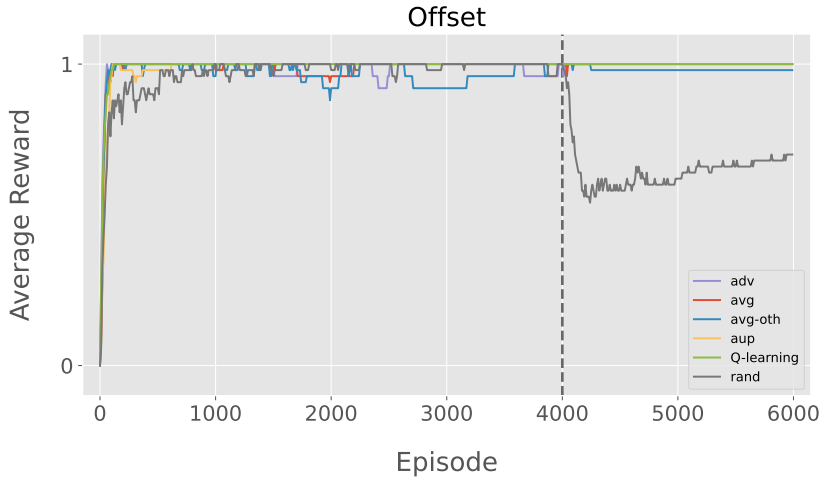


Fig. 5: Average reward for different approaches in the `Offset` environment. The reward is averaged per time step over 50 trials ($\varnothing \in \mathcal{A}$). None of the approaches, except the random-action baseline, show offsetting behavior, where the box is saved first but then put on the conveyor belt again.

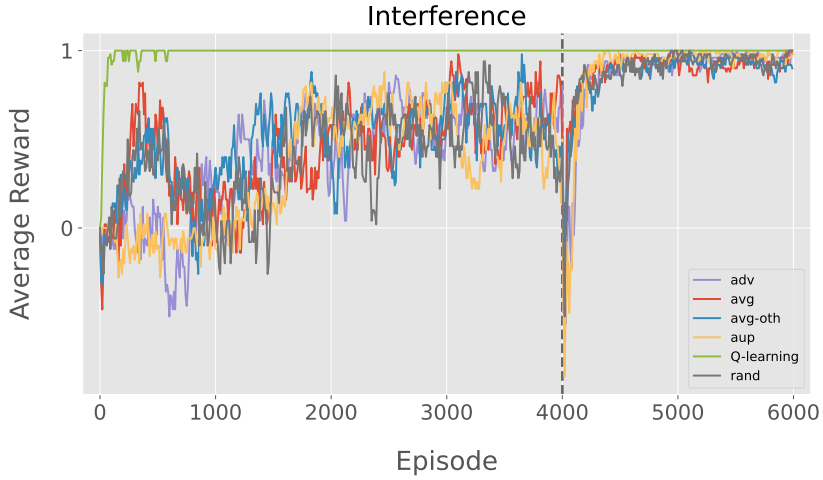


Fig. 6: Average reward for different approaches in the `Interference` environment. The reward is averaged per time step over 50 trials ($\varnothing \in \mathcal{A}$). All of the methods show near-optimal performance in the end, indicating that the agent does little or not interfere with the moving orange pallet.

6.2 Dropping the no-op action

Figures 7 to 10 show the results in the modified environments where the no-op action is excluded from the action space. These results show that `Options` still imposes a challenge to all approaches, while all baselines, except standard Q-Learning, manage to avoid side-effects in `Damage`.

None of the approaches show neither offsetting nor interfering behavior, while all baselines except the random-action baseline, show correcting behavior. Again this is most likely due to the delayed side-effect in this environment.

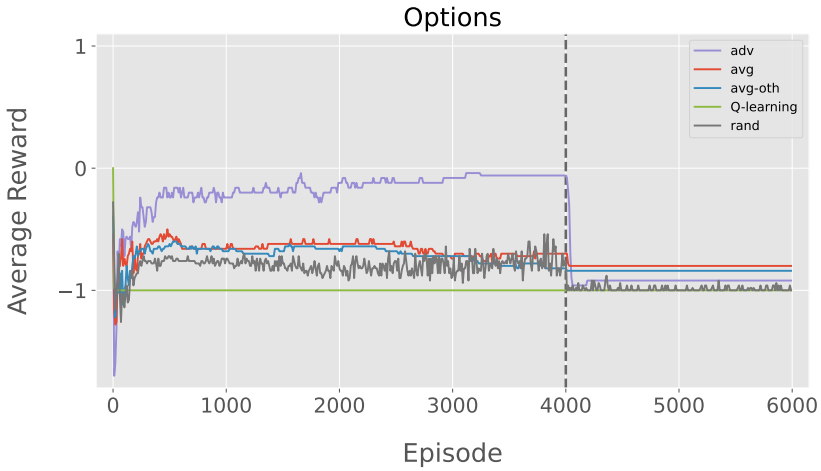


Fig. 7: Average reward for different approaches in the `Options` environment. The reward is averaged per time step over 50 trials ($\varnothing \notin \mathcal{A}$). All methods show a clear performance drop after the exploration switch.

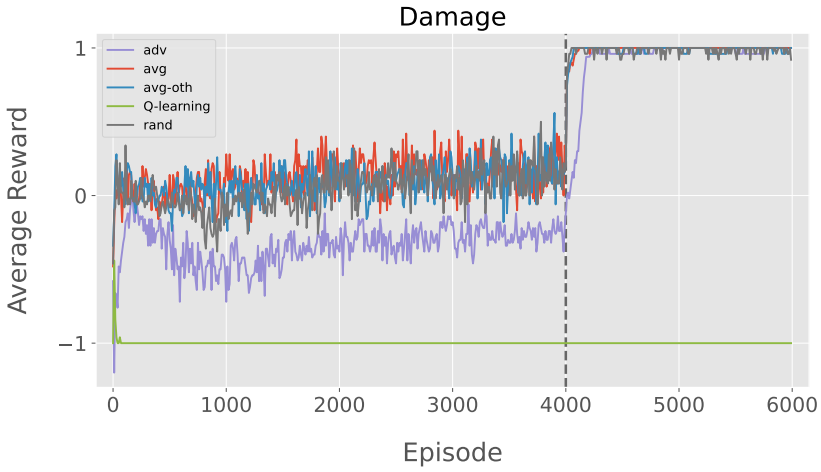


Fig. 8: Average reward for different approaches in the `Damage` environment. The reward is averaged per time step over 50 trials ($\varnothing \notin \mathcal{A}$). All methods, except Q-Learning, show near-optimal performance after the exploration switch.

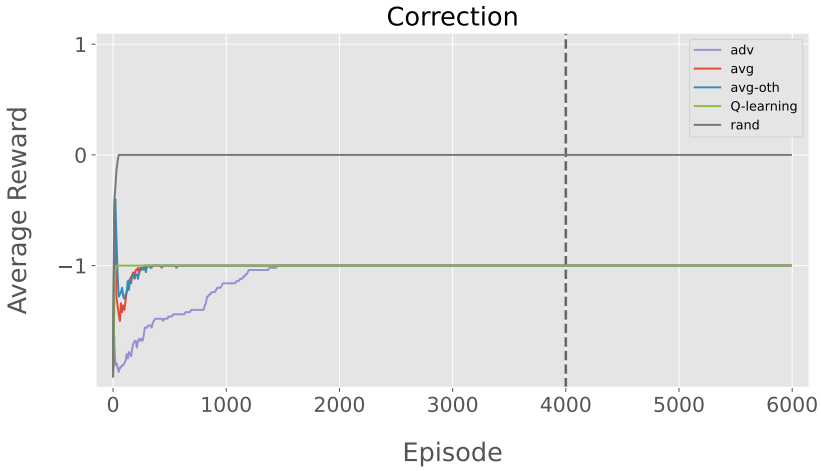


Fig. 9: Average reward for different approaches in the `Correction` environment. The reward is averaged per time step over 50 trials ($\varnothing \notin \mathcal{A}$). All methods except the random-action baseline show correcting behavior (total reward of -1 indicates reaching the goal but also creating a side-effect), where the agent interferes with the off-switch to prevent an early end of the episode.

Fig. 10: Average reward for different approaches in the `Offset` environment. The reward is averaged per time step over 50 trials ($\varnothing \notin \mathcal{A}$). None of the approaches show clear offsetting behavior, where the box is saved first but then put on the conveyor belt again.

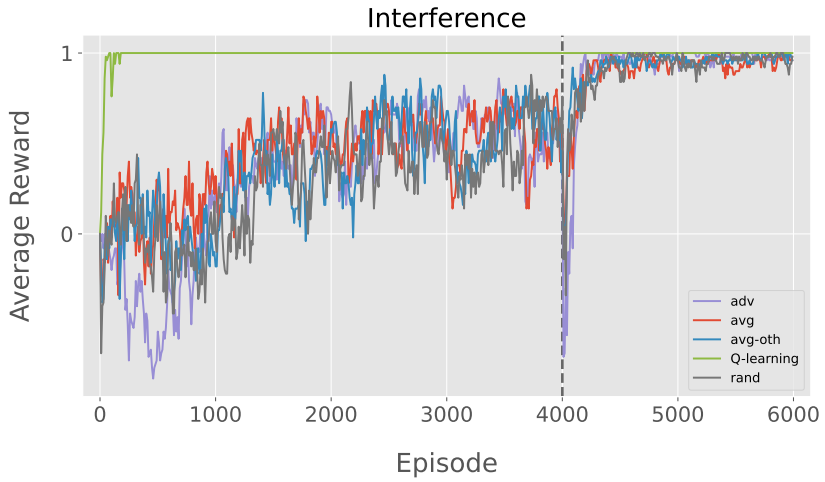


Fig. 11: Average reward for different approaches in the `Interference` environment. The reward is averaged per time step over 50 trials ($\varnothing \notin \mathcal{A}$). All of the methods show near-optimal performance in the end, indicating that the agent does little or not interfere with the moving orange pallet.

7 Discussion

The poor performance in `Options` indicates, that the advantage, average and average-others baselines struggle with environments that require irreversible actions to be taken. Especially in comparison with `Damage`, where each state can be reached again and all approaches achieve optimal performance. Also a comparison between fig. 2 and fig. 7 shows a visible difference in final performance, indicating that the missing no-op action has a performance impact when irreversible actions are required to achieve the goal. This suggests that our approaches are indifferent to which part of the environment is irreversible.

Overall, the average and average-others baselines perform very similarly. This indicates that it is not relevant whether the selected action is part of the average in the penalty term or not. This is probably due to the action value of the selected action not being a particular outlier compared to the average of all action values in the given state.

The advantage baseline was capable of outperforming model-free AUP in `Options` during the first 4,000 episodes with $\epsilon = 0.8$. Moreover, we find that the performance of the average and average-others baseline is better compared to the advantage baseline and model-free AUP in the `Damage` environment during the exploration phase. However, all approaches rise to the optimal performance once the exploration switch is reached. We assume this phenomenon has a connection to the amount of "free space" available to the agent until it comes in contact with the side-effect. While in `Options` this is rather soon, as the side-effect is just one action away from the initial state and 3 fields around the box are available for the agent, in `Damage` the side-effect is two actions away and the agent has 6 fields that are uninvolved by the human.

Unsurprisingly, the advantage, average and average-others baselines show correcting behavior, meaning they intervene with the off-switch in `Correction`. The agents using these baselines, have learned that only after this intervention they are capable of reaching the goal state. Our approaches are incapable of avoiding this side-effect as it comes with a time-delay (the episode still continuing after two steps) and by design our approaches cannot handle such side-effects, as does model-free AUP. AUP is supposed to handle delayed side-effects, however only side-effects, that are delayed by 8 time steps. Interestingly, the random-action baseline manages to prevent correcting behavior in `Correction`, which requires further investigation.

8 Conclusion

We propose three different, alternative baselines to attainable utility preservation that do not build upon a no-op action, which induce safer, yet effective behavior than standard Q-Learning. We evaluate all three baselines on two separate versions of five AI safety Gridworlds comparing them to model-free AUP, Q-learning and a random baseline. Our proposed baselines require less assumptions and therefore are more broadly usable, but also show less side-effect avoiding potential in environments with irreversible actions and are more sensitivity to parameters.

8.1 Future Work

We suggest future work on investigating the performance of the proposed baselines in larger, more complex and multi-task environments, as well as in environments with larger action spaces, to determine the extent to which our proposed baselines induce safe and effective behavior. Also coping with delayed side-effects in unspecified time frames is still an open challenge to be solved.

9 Acknowledgements

This research was funded in whole, or in part, by the Austrian Science Fund (FWF) [P 33656-N]. For the purpose of open access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Alamdari, P.A., Klassen, T.Q., Icarte, R.T., McIlraith, S.A.: Be Considerate: Objectives, Side Effects, and Deciding How to Act, http://arxiv.org/abs/2106.02617
2. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete Problems in AI Safety, http://arxiv.org/abs/1606.06565
3. Armstrong, S., Levinstein, B.: Low Impact Artificial Intelligences. https://doi.org/10.48550/arXiv.1705.10720, http://arxiv.org/abs/1705.10720
4. Bellemare, M.G., Candido, S., Castro, P.S., Gong, J., Machado, M.C., Moitra, S., Ponda, S.S., Wang, Z.: Autonomous navigation of stratospheric balloons using reinforcement learning. Nature **588**(7836), 77–82 (2020)
5. Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep Reinforcement Learning from Human Preferences. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html
6. Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., R Ruiz, F.J., Schrittwieser, J., Swirszcz, G., et al.: Discovering faster matrix multiplication algorithms with reinforcement learning. Nature **610**(7930), 47–53 (2022)
7. Gleave, A., Toyer, S.: A Primer on Maximum Causal Entropy Inverse Reinforcement Learning. https://doi.org/10.48550/arXiv.2203.11409, http://arxiv.org/abs/2203.11409
8. Krakovna, V., Orseau, L., Martic, M., Legg, S.: Penalizing Side Effects using Stepwise Relative Reachability. In: Espinoza, H., Yu, H., Huang, X., Lecue, F., Chen, C., Hernández-Orallo, J., hÉigeartaigh, S.O., Mallah, R. (eds.) Proceedings of the Workshop on Artificial Intelligence Safety 2019. CEUR Workshop Proceedings, vol. 2419. CEUR (2019), http://ceur-ws.org/Vol-2419/#paper1
9. Krakovna, V., Orseau, L., Ngo, R., Martic, M., Legg, S.: Avoiding Side Effects By Considering Future Tasks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 19064–19074. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/dc1913d422398c25c5f0b81cab94cc87-Paper.pdf
10. Leech, G., Kubicki, K., Cooper, J., McGrath, T.: Preventing Side-effects in Gridworlds, https://www.gleech.org/grids
11. Leike, J., Martic, M., Krakovna, V., Ortega, P.A., Everitt, T., Lefrancq, A., Orseau, L., Legg, S.: AI Safety Gridworlds. https://doi.org/10.48550/arXiv.1711.09883, http://arxiv.org/abs/1711.09883
12. OpenAI, Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P.a., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., Pinto, H.P.d.O., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., Zhang, S.: Dota 2 with Large Scale Deep Reinforcement Learning. https://doi.org/10.48550/arXiv.1912.06680, http://arxiv.org/abs/1912.06680
13. Saunders, W., Sastry, G., Stuhlmüller, A., Evans, O.: Trial without Error: Towards Safe Reinforcement Learning via Human Intervention. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. pp. 2067–2069. AAMAS '18, International Foundation for Autonomous Agents and Multiagent Systems (2018)

14. Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., Silver, D.: Mastering Atari, Go, chess and shogi by planning with a learned model **588**(7839), 604–609 (2020). https://doi.org/10.1038/s41586-020-03051-4, https://www.nature.com/articles/s41586-020-03051-4

15. Shah, R., Krasheninnikov, D., Alexander, J., Abbeel, P., Dragan, A.: Preferences Implicit in the State of the World (2022), https://openreview.net/forum?id=rkevMnRqYQ

16. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search **529**(7587), 484–489 (2016). https://doi.org/10.1038/nature16961, http://www.nature.com/articles/nature16961

17. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. Adaptive Computation and Machine Learning Series, The MIT Press, second edition edn. (2018), http://incompleteideas.net/book/the-book.html

18. Turner, A.M., Hadfield-Menell, D., Tadepalli, P.: Conservative Agency via Attainable Utility Preservation. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. pp. 385–391. AIES '20, Association for Computing Machinery (2020). https://doi.org/10.1145/3375627.3375851

19. Turner, A.M., Ratzlaff, N., Tadepalli, P.: Avoiding Side Effects in Complex Environments. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H.T. (eds.) Advances in Neural Information Processing Systems. NeurIPS 2020, vol. 33, pp. 21406–21415. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/hash/f50a6c02a3fc5a3a5d4d9391f05f3efc-Abstract.html

20. Turner, A.M., Smith, L.R., Shah, R., Critch, A., Tadepalli, P.: Optimal Policies Tend To Seek Power (2021), https://openreview.net/forum?id=l7-DBWawSZH

21. Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature **575**(7782), 350–354 (2019)

Supplementary Material

The code to reproduce the results as well as the requirements to setup the experiments are published on GitHub[5]. This repository also contains raw data for all conducted plots as well as logged primary and auxiliary action-values functions of all introduced baselines.

A Parameter Study

This chapter shows results on experiments with varying parameters, demonstrating the sensitivity of the proposed baselines to these parameters. These results show the raw outcome tallies for proposed baselines in all tested environments. Agents were evaluated using the parameter ranges as shown in table Table 2. Result figures show the outcome over 50 trials.

γ	0.875	0.938	0.969	0.984	0.992	0.996	0.998	0.999			
λ	0.36	0.4	0.5	0.6	0.7	0.8	1.1	1.7	3.3	1000.0	
$\|\mathcal{R}\|$	0	5	10	15	20	25	30	35	40	45	50

Table 2: Considered parameters for the parameter study, for the discount factor γ, the penalty scaling factor λ and the amount of auxiliary reward functions $|\mathcal{R}|$.

A.1 Including the no-op actions

The results in Figures 12 to 14 were conducted in environment variants that include the no-op action.

[5] `https://github.com/fkabs/attainable-utility-preservation`

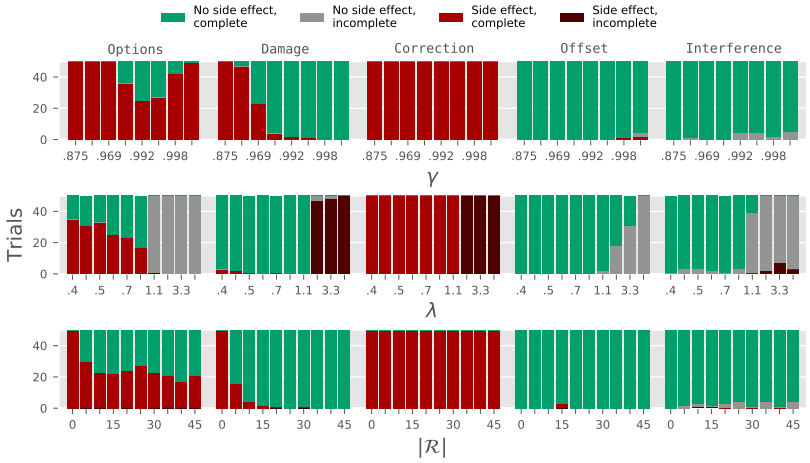


Fig. 12: Count plot of the advantage baseline, that shows outcome tallies across a range of parameter settings for all five environments ($\varnothing \in \mathcal{A}$).

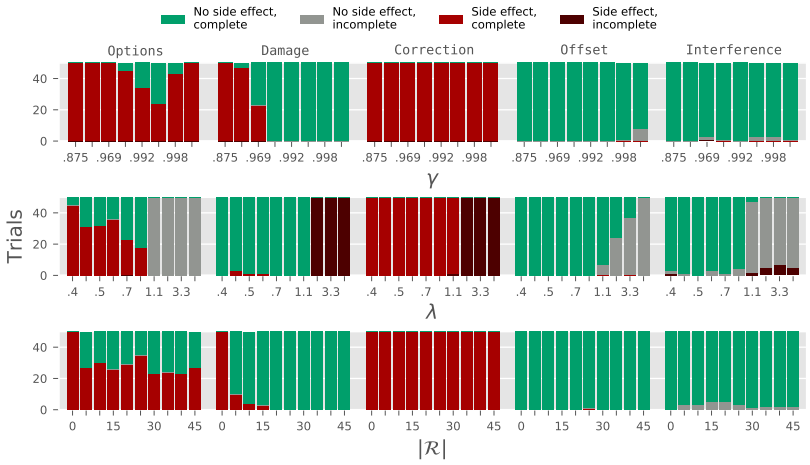


Fig. 13: Count plot of the average baseline, that shows outcome tallies across a range of parameter settings for all five environments ($\varnothing \in \mathcal{A}$).

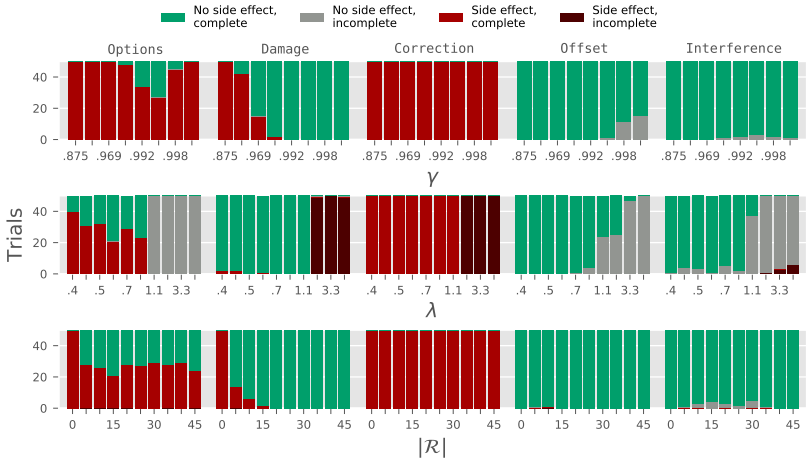


Fig. 14: Count plot of the average-others baseline, that shows outcome tallies across a range of parameter settings for all five environments ($\varnothing \in \mathcal{A}$).

A.2 Excluding the no-op action

The results in figs. 15 to 17 were conducted in environment variants that exclude the no-op action.

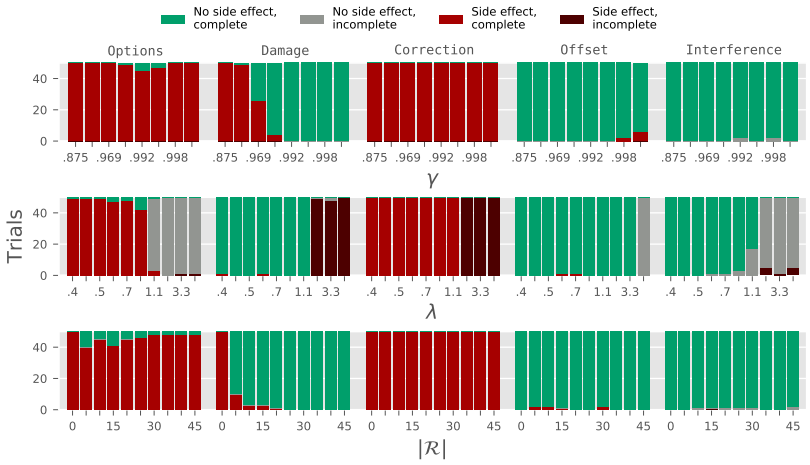


Fig. 15: Count plot of the advantage baseline, that shows outcome tallies across a range of parameter settings for all five environments ($\varnothing \notin \mathcal{A}$).

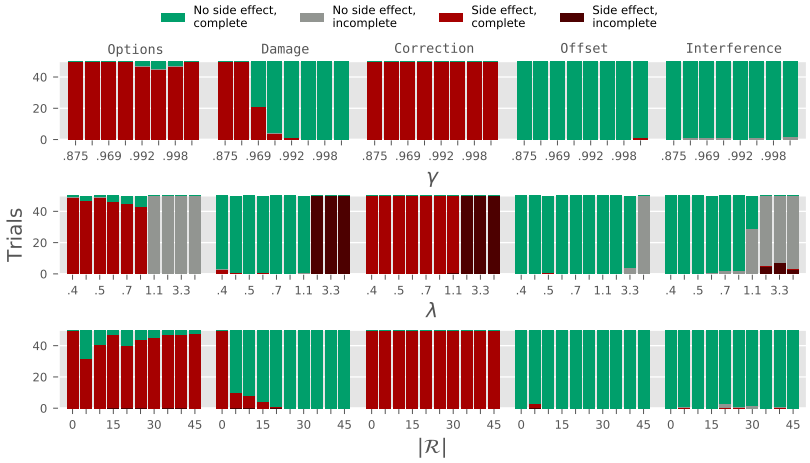


Fig. 16: Count plot of the average baseline, that shows outcome tallies across a range of parameter settings for all five environments ($\varnothing \notin \mathcal{A}$).

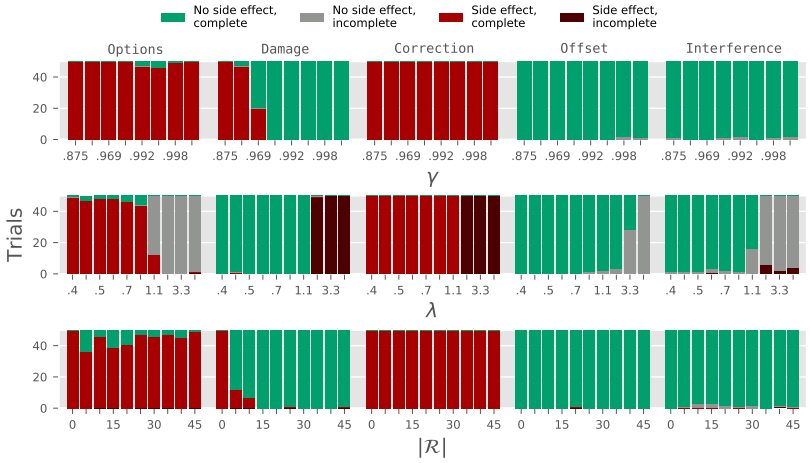


Fig. 17: Count plot of the average-others baseline, that shows outcome tallies across a range of parameter settings for all five environments ($\varnothing \notin \mathcal{A}$).