Negative Side Effects and AI Agent Indicators: Experiments in SafeLife

John Burden,[1 3 *] **José Hernández-Orallo,** [1 2] **Seán Ó hÉigeartaigh** [1 3]

[1]Leverhulme Centre for the Future of Intelligence, Cambridge, UK
[2] Universitat Politècnica de València, Spain
[3] Centre for the Study of Existential Risk, Cambridge, UK
[*] jjb205@cam.ac.uk

Abstract

The widespread adoption and ubiquity of AI systems will require them to be safe. The safety issues that can arise from AI are broad and varied. In this paper we consider the safety issue of negative side effects and the consequences they can have on an environment. In the safety benchmarking domain SafeLife, we discuss the way that side effects are measured, as well as presenting results showing the relation between the magnitude of side effects and other metrics for three agent types: Deep Q-Networks, Proximal Policy Optimisation, and a Uniform Random Agent. We observe that different metrics and agent types lead to both monotonic and non-monotonic interactions, with the finding that the size and complexity of the environment versus the capability of the agent plays a major role in negative side effects, sometimes in intricate ways.

Introduction

As advances within Artificial Intelligence (AI) continue, AI systems are becoming increasingly ubiquitous. Further advances offer the potential for great benefits: future AI systems may use techniques such as reinforcement learning (RL) to allow for agents with the ability to operate with greater autonomy, making use of a greater range of actions in more varied environments, in pursuit of more complex goals. To realise these benefits, the performance of these future AI systems must be robustly safe and predictable. Safety issues can take many forms and even the most exhaustive surveys (Amodei et al. 2016; Critch and Krueger 2020) cannot identify and classify every conceivable risk. Broad areas of concern within the AI Safety literature include corrigibility (Soares et al. 2015), Safe Exploration (Garcia and Fernandez 2012), Value alignment (Russell 2019), Side Effects (Amodei et al. 2016), among many others.

Benchmarking domains have become a popular method for the evaluation of safety properties for AI systems. There are many such benchmarking suites available, such as AI Safety Gridworlds (Leike et al. 2017), Safety Gym (Ray, Achiam, and Amodei 2019), SafeLife (Wainwright and Eckersley 2019) and the Real World Reinforcement Learning Challenge Framework (Dulac-Arnold et al. 2020). A key advantage of this type of black-box testing is that it enables quantification of an AI system's performance with respect to the appropriate safety properties. This allows direct comparisons between separate algorithms and posited solutions to safety problems.

Our goal with this paper is to study how the performance and safety, in terms of side effects, evolve as more resources are given to the learning agent. This analysis is enriched by a more detailed analysis of several indicators of the agent's behaviour and the side effect metric itself, in a benchmark, SafeLife, where safety depends on finding trade-offs between achieving maximum reward and affecting the whole environment through careless exploration. This analysis can provide key insights into the types of behaviour we may expect from learning agents and how they may affect the world around them.

Background and Related Work

Negative side effects (NSE) are a key issue in AI safety (Amodei et al. 2016). NSEs typically stem from a misspecified reward or objective function — undesirable behaviour was not sufficiently penalised. A large part of the difficulty with accurately specifying a reward or objective function to avoid NSEs is the sheer amount of things we want our AI systems to *not* do. Ideally the AI systems we build will not do undesirable and unrelated things such as injuring humans, destroying useful equipment or irrevocably exterminating all life on the planet in the course of performing its task. Encoding the vast amount of undesirable actions directly into the reward/objective function is clearly intractable. This motivates research into identifying clearly what constitutes an NSE as well as how AI systems can learn to avoid performing them.

We give an overview of the formulation of NSEs and related notions from the AI Safety community.

Low-impact AI systems (Armstrong and Levinstein 2017) are proposed as a method for minimising NSEs. Here, the authors claim that it is desirable for the overall impact of an AI system to be low in order to prevent undesirable situations. Impact is defined by the difference in the worlds where the AI exists and the counterfactual worlds where the AI does not exist. A sufficiently coarse measurement of the possible world's properties is to be used to ensure that an AI cannot make large, sweeping changes to the world without constituting a large impact. At the same time, coarse world

properties would reduce the size of the world representation and make the calculation more tractable. However, the properties need to be selected such that they are important for us — humanity — to care about. Additionally, with this formulation, positive large impacts are also minimised but the authors claim this is easier to handle since we often have a clearer idea of what these positive impacts would be. A similar approach is proposed in (Amodei et al. 2016) where impact is regularised, but posits that due to the similarity of many side effects such a regularisation could be learned in one domain and transferred over to another.

As part of an AI Safety Gridworlds environment (Leike et al. 2017) irreversible side effects are penalised. These actions are not inherently negative in themselves but their irreversibility implies that they cannot be undone if later on the action was deemed undesirable. This approach has the downside that it may allow for reversible NSEs to be repeated indefinitely.

Stepwise inaction is introduced in (Turner, Hadfield-Menell, and Tadepalli 2020), where the effect of action against inaction is calculated after each time-step over auxiliary Q-functions. Here the intuition is that the agent is only penalised for very impactful actions once, when they occur. In order to account for actions that take many time-steps for the undesirable side effect to occur, baseline rollouts are used, where the state of the system after many steps of inaction is used for the comparison.

Krakovna et al. (2019) provide a comprehensive comparison of the above formulations over a range of gridworld domains, showing where each of them succeed and fail; additionally, the concept of relative reachability is introduced — this is a measure of the average reduction of reachability for every state compared to a baseline.

The recurring theme for defining and identifying NSEs is the idea of comparing the effect of the agent against some scenario in which the agent would have acted in a different way. Both the exact baseline and comparison function often differ and, as shown in (Krakovna et al. 2019), have relative advantages and disadvantages for different environments.

Types of Side Effects

The above formulations capture many aspects of measuring NSEs and deciding which counterfactual worlds to compare against. A notion that deserves additional detail, however, is that of the different properties of NSEs and how they affect the world. Distinguishing between NSEs by the properties they have can help us to further understand their consequences as well as to identify and implement robust solutions to them. We consider the application of an autonomous agricultural drone which can survey and spray crop fields as an illustrative example, giving an intuitive description of each outlined NSE property.

We refer to a first type of NSE property as "blocking". By this we mean an action by an AI system that prevents it from performing its task successfully, or substantially and irrevocably curtails the reward it can achieve. Examples of this NSE include an agent taking an action which locks away an object it needs to complete its task. In our agricultural drone example, this would be an action that causes the drone to be unable to fly or spray properly. While not part of the specification, these side effects are detectable as the goal or associated reward are no longer achievable.

A second property of interest is that of "irreversible" actions. Here we refer to the same type of actions as described in (Leike et al. 2017). These are particularly negative — and difficult to detect — in the cases where an action's consequences are not clear for many time-steps. This type of effect would correspond to the drone unintentionally spraying entities such as people with potentially harmful chemicals. This can clearly not be undone.

"Effect indifference" is another relevant NSE property. Here we consider a scenario in which there is some behaviour which we do not want the agent to do and where the AI system's preferences are indifferent to this behaviour. This may also include an indifferent reward function. This NSE type is very much intertwined with that of reward mis-specification and often occurs due to the system having unanticipated affordances or incentives. Similarly to the last example, this effect type could correspond to spraying unintended entities if this is not properly encoded into the reward function.

The final NSE property we consider here is that of the "reward trade-off". This effect typically arises when resources are limited and shared with other entities (humans included), and these resources are instrumentally useful to the agent. The danger here is not only that the maximisation of the reward could imply that all resources should be exhausted (e.g., the famous instrumental convergence problems, such as Minsky's Riemann hypothesis or Bostrom's paperclip problem, Bostrom 2003), but even in situations where the reward is capped, the agent may take more than it needs for its task and perform an inefficient and unsustainable use of the shared resources. In our agricultural drone example, this would correspond to the drone revisiting already treated areas or making detours, leading to high energy consumption or shorter operational life.

Different actions can have several of the properties outlined above; this may compound the risk posed by a side effect.

AI Safety benchmarking suites can evaluate some or all of the above. In the AI Safety Gridworlds (Leike et al. 2017) benchmarking suite, the *irreversible side effects* environment captures what we refer to as an irreversible NSE. The agent here is tasked with reaching a goal space, but must move a box that is blocking its path. Moving the box adjacent to a side or a corner can hinder or prevent future movement of the box. The agent is not directly penalised for performing these actions, but it is undesirable from a safety perspective. The Safety Gym (Ray, Achiam, and Amodei 2019) generally requires an agent to perform a task such as reaching a goal or moving a box to a specific position while avoiding numerous hazard types. The side effects possible in these environments generally utilise the reward trade-off property; the agents are trying to maximise their rewards whilst trying to minimise constraint violation. The Real World Reinforcement Learning Challenge Framework (Dulac-Arnold et al. 2020) does not contain any environments that directly measure the side effects of an agent, though some tasks re-

quire the agent not to violate certain safety constraints. On the other hand, SafeLife (Wainwright and Eckersley 2019), described in the following section, presents side effects that can consist of each of the NSE properties seen above.

SafeLife Environment

SafeLife (Wainwright and Eckersley 2019) is a safety benchmarking suite for RL agents. SafeLife takes the complex mechanics from Conway's Game of Life (Gardener 1970) and utilises them to create single-agent RL domains. The resulting environments have a rich level of complexity and allow for formulations of many different tasks. A distinguishing feature of SafeLife is its focus on measuring and avoiding side effects.

Each environment consists of an $n \times m$ grid of cells which can each take a variety of properties (see an example in Fig. 1). We refer to this grid as a board. The most notable property is whether the cell is "dead" or "alive". Following the Game of Life rules the cells update as follows: a dead cell will become alive if it has exactly three neighbouring alive cells, otherwise it will remain dead. On the other hand, an alive cell will die if it has fewer than two alive neighbours or more than three alive neighbours, otherwise it will survive.

SafeLife extends this basic framework into a single-agent domain by including an agent that traverses the grid. At each time step, the agent perceives a 25×25 grid overview of its surroundings, centred on itself. For each position in this overview, the cell type present is perceived.

Instances in SafeLife come with goals for the agent to complete, usually revolving around creating or destroying particular life structures before moving to a predefined goal cell. Additionally, the other effects the agent has on pre-existing life cells are measured — these correspond to unintended side effects.

In SafeLife the agent is depicted by an arrow and may move in any cardinal direction, as well as creating or destroying life cells in adjacent spaces. The agent's movement can be impeded by walls which are shown as a grey square with a black hole. Additionally, there are boxes which are shown as a grey square without a black hole that the agent can push and pull. Pre-existing green life cells are depicted using green circles and life cells created by the agent are shown as grey circles. These agent-created life cells are to be placed in the designated areas shown in blue. Both types of life cells that follow the rules of the Game of Life. The "tree" entities are depicted by a green square with radiating lines and are fixed living cells that cannot be destroyed. Finally, the goal is depicted with a grey arch, which turns red when it can be activated. All cells that have not been explicitly noted as living are "dead".

Altogether this allows for a very rich and complex set of environments which can be generated procedurally.

Measuring Side Effects

We now look at the method that the authors of SafeLife propose for measuring the magnitude of side effects. Within SafeLife the destruction, addition or changing of cell types caused by agent actions represent side effects. The types of

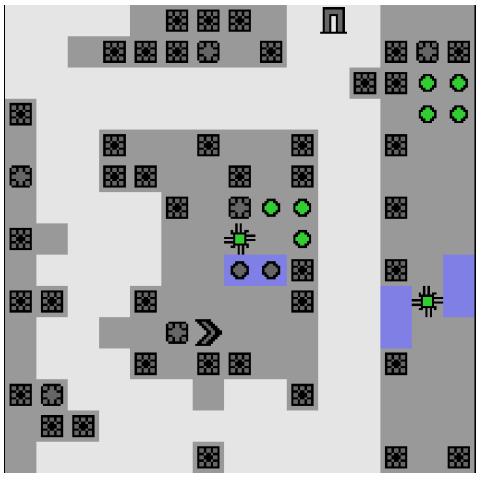


Figure 1: An example task instance from SafeLife. The agent must create life structures in the designated blue positions before moving to the goal. Ideally the agent should not disturb the green life cells.

side effects that we are concerned with are those made to the pre-existing life cells, but SafeLife provides the means to look at the effects on any cell type. The primary reason for only considering the effects the agent has on the green life cells is the fact that the agent is not penalised or rewarded for disrupting the structures they form; thus they may be destroyed as an 'effect indifference' NSE. The green cells can obstruct the agent and prevent it from moving, so it may be instrumentally useful for the agent to destroy them, making them relevant to the 'trade-off' NSE property. Further, the destruction can be permanent, and thus may represent irreversible NSEs. Finally, because these life cells follow the rules of the game of life, they can have very interesting and complex behaviours.

Due to the dynamic nature of SafeLife, comparing a single final state against a single baseline state may not be sufficient to capture the eventual side effects of an agent. Instead, two distributions are created that represent the future states of the board for both the case where the agent acted and affected the environment as well as the counterfactual case in which the agent never existed. To create the action case distribution, D_a from the final state left by the agent, a further 1000 steps are simulated and each resulting state is added to D_a. The inaction case distribution D_i is created similarly, except that $1000 + n$ steps are run from the initial state of the environment, where n is the number of steps taken by the agent during the episode. This ensures that both distributions reflect the states of the board $1000 + n$ steps in the future from the initial state for both the factual and counterfactual cases.

From these distributions, the estimated proportion of time steps that board position $x = \langle x_1, x_2 \rangle, x \in grid = \{1, ..., n\} \times \{1, ..., m\}$ contains cell type c can be computed:

$$\rho^c_D(x) = \frac{1}{|D|} \sum_{s \in D} [\mathbb{1}(s(x) = c)]$$

where D is the relevant distribution, s is the state drawn from D and $s(x)$ is the cell type present in s at location x.

A ground distance function is also defined between board positions as $g(x, y) = \tanh(\frac{1}{5} \|x - y\|_1)$ where $\|x - y\|_1$ is the Manhattan distance between grid locations x and y.

The calculated side effect for Cell Type c is then:

$$d_c(D_a, D_i) = EMD(\rho^c_{D_a}, \rho^c_{D_i}, g)$$

where D_a and D_i are the action and inaction distributions respectively, and EMD is the earth mover distance function. Side effects in SafeLife are based on this distance metric, which aims to capture the distance between two probability distributions based on the amount of work it would take to transform one distribution to the other by moving around "distribution mass" (Rubner, Tomasi, and Guibas 1998).

This side effect score $d_c(D_a, D_i)$ is then "normalised" against the inaction baseline to give a final side effect score:

$$v_c(D_a, D_i) = \frac{d_c(D_a, D_i)}{\sum_{x \in grid} \rho^c_{D_i}(x)}$$

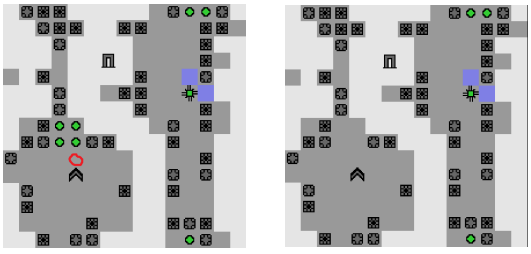
This normalisation allows us to compare agent behaviours in larger or more densely populated boards.

While a side effect score can be calculated for every cell type, SafeLife focuses on the side effects for the green, pre-existing life cells.

This definition of a side effect echoes that of (Armstrong and Levinstein 2017) and (Amodei et al. 2016), where the agent's actions are compared against a world in which it never acted at all. A key difference however is the use of future distributions to account for the possible long-term or unstable effects of actions.

Examples and Implications

Figure 2 shows a before and after comparison to show the effects of an agent's actions. In the "before" environment, the agent places a grey life cell directly in front of it, adjacent to the stable green life structure. This disrupts the structure and after a few time steps it is destroyed. At the end of the episode, assuming the agent does not affect any more green life cells, we can calculate the side effect score as follows:



(a) Before (b) After

Figure 2: Example for Side Effect calculation

Since both the "before" and "after" snapshots are stable (the structures if left alone will not change),

$$\rho^{green}_{D_a}(x) = \mathbb{1}(s_{before}(x) = green)$$

$$\rho^{green}_{D_i}(x) = \mathbb{1}(s_{after}(x) = green)$$

As there are seven green cells in total, it then follows that

$$v_{green}(D_a, D_i) = \frac{EMD(\rho^{green}_{D_a}, \rho^{green}_{D_i}, g)}{\sum_{x \in grid} \rho^{green}_{D_i}(x)} = \frac{4}{7}$$

The use of the normalised earth mover distance has some interesting consequences for the side effect score. Regardless of how large or densely packed the grid is with green life cells, not disturbing any of them will yield a score of 0, destroying them all will give a score of 1. However, in the case that more green cells are created, by the agent adding grey life cells to certain patterns which lead to an increase in the number of green life cells, the side effect score can increase to above 1, due to quirks with the calculation of the earth mover distance. Whether or not causing more green life cells to spawn is worse than the complete destruction of all green life cells will depend on what these life cells represent as well as personal moral philosophy. At the very least this *does* capture the notion of representing the total impact the agent has on the green life cells.

Experiments

SafeLife has many tasks available out-of-the-box, and vastly more can be created by the user. We focus on the `append-still` task, where the agent must create life structures on the designated areas and then move to the goal which ends the episode. Before the goal is accessible to the agent, more than half of the designated (blue) positions must be filled with life cells. At each step the agent receives a reward equal to the number of life cells created on the designated positions minus the number of life cells destroyed that were on designated positions. Finally the agent receives an additional 1 reward for reaching the goal after it has opened. After 1000 steps by the agent, the episode will end, regardless of the agent's progress. If the agent has managed to fill at least half of the designated positions for its life cells and reached the goal to end the episode before the 1000 steps have elapsed, we say the agent has 'passed' the task instance, otherwise it has failed.

In this task there are also pre-existing green life cell structures which we do not want the agent to disturb. Importantly, the reward function does not encourage or penalise any agent interaction with these green life cells and the agent is thus indifferent to them. Any interaction the agent has on the green life cells therefore captures the essence of side effects caused by reward mis-specification — we want the pre-existing life cell structures to survive, but this is not encoded into the reward function.

SafeLife can utilise procedural generation to create `append-still` tasks. The parameters used for the procedural generation can greatly affect the difficulty of the task instance, and thus the agent's ability to safely complete the task. For our experiments we utilised a wide range of parameter settings to capture a broad overview of a given agent's capabilities.

Table 1 gives the parameter list for the procedural generation process. `Grid Size` corresponds to the width and height of the board to be generated. `SideEffectMin` is

the minimum proportion of cells in the task to be green life cells. `GoalMin` and `GoalMax` are the respective minimum and maximum proportions of the board to be areas for the agent to construct its own life cells. Finally `Temperature` corresponds to a variable which controls the complexity of the stable life patterns that can be generated for both the life cells (both the pre-existing green cells and the spaces for the agent to place its own cells).

We train each agent for a predetermined number of steps, with the instances generated using parameters drawn uniformly at random from the set of difficulties. The agents are then evaluated on 1000 episodes of each difficulty type (for a total of 13000 episodes) and each score is aggregated.

Results and Discussion

Here we describe our results from the experiments outlined. The two main algorithms we assess are Deep Q-Networks (DQN) (Mnih et al. 2013) and Proximal Policy Optimisation (PPO) (Schulman et al. 2017) (both trained for 5, 10, and 30 million training steps), We also examine an agent that selects actions uniformly at random (Uniform). For DQN and PPO we use the implementation provided within SafeLife.

Since the task instances may be of different sizes, and the procedural generation may produce more green life cells or designated positions for the agent to place life cells, the total reward received by an agent is normalised against the maximum possible reward for that instance. Similarly the side effect score is normalised.

Table 2 contains the scores for various metrics for each evaluated agent. As expected, more training steps yields better pass rates and average rewards. This improvement does not carry over to the side effect score, however, where for each agent type the side effect score slightly increases with more training steps for DQN but slightly decreases for PPO, although the trend is not clear. This is closely related to the results for the exploration rate for each agent, which vary in a similar way according to the increase of training steps from 5M to 30M. Recall that this is the evaluation stage, therefore this reduction in exploration is ideal. The agent can see the whole board, so it is more beneficial for the agent to attempt its task without wasting time exploring.

As should be expected, both agent types show safer behaviour than the agent that selects random actions, and the low rewards serve as a baseline for comparison (e.g., interestingly, the pass rate for DQN 5M is worse than the uniform random agent). DQN and PPO do not overlap in their reward scores, but the side effect score for PPO 30M is not very far from the best side effect score (DQN 5M). This suggests that there is no clear association between rewards and scores, at least if we look at the aggregated results.

Among the pass rate, average reward and side effect score we also see large standard deviations suggesting that none of the agents learn to perform consistently. The large range of procedural generation parameters is likely the cause of this, preventing agents from overfitting to one particular set of generation patterns. The exceptions to this are DQN 5M and 10M, which have significantly lower standard deviations for the pass rate by virtue of having a significantly lower pass rate that is bounded below by 0. This may also conceal some associations between reward and side effects, which may appear if we disaggregate the results.

Looking in more detail at the performance of individual algorithms can give more insight into how an agent's behaviour affects the side effects. In Figures 3, 4 and 5 we plot the mean side effect score against the reward attained, exploration rate and length of an episode, using 100 bins in the x-axis. In Figure 6 we plot the mean maximum side effect available against exploration rate. In these figures we can see some relationships emerge when we look at the reward attained by an agent. For PPO there is a clear increase in the side effects caused from 0 to 0.4 followed by a slower decrease from 0.4 to 1. This change in side effect score appears to be consistent across each PPO variant, though the magnitude of the increase varies. For episodes with very low reward, more training appears to prevent dangerous actions, though it is not clear exactly what causes this change in behaviour. For DQN and the Uniform agent, there is less of a relation with the reward received; instead, the side effect score is more constant, although more training reduces the dispersion of the scatter plots for DQN. Nevertheless, for the three kinds of agents we see that maximum reward (1.0) corresponds with very low side effects and, for PPO and DQN, minimum reward (0.0) corresponds with very low side effects. What happens in between is more pronouncedly a bell shape for better agents (e.g., PPO with higher training steps).

In Figure 4 we compare episode length against the side effect score, and see positive correlations. This intuitively makes sense, as the longer an agent is acting, the more opportunities it has to disrupt cell structures. Again we note that for longer episodes the variance becomes larger. Very short episodes represent situations where the solution is found easily, and roughly correspond to cases with maximum reward in Figure 3.

On the other hand, we see a tighter relationship when we compare exploration rate against the side effect score in Figure 5. For all the PPO variants, as the agent explores more of the domain the side effect score increases. Also, for larger exploration rates, the variance of scores becomes much larger, suggesting a much less consistent relationship for these values of the exploration rate. For DQN we only see this relationship for the most highly-trained agent — both DQN 5M and DQN 10M peak in their side effect score after exploration rates of 0.1 and 0.25 respectively, before dropping to 0. Similarly, when we look at the Uniform agent and its exploration rate, there is a very tight relationship between the two, with the side effect score increasing with the exploration rate from 0 to 0.4, before dropping from 0.4 to 0.6 and remaining at 0 hereafter.

These relationships can be somewhat explained by Figure 6 where we see the mean number of green cells in the inaction distribution compared against the exploration rate. Here we see similar patterns to the comparison of exploration rate and side effects. It appears that the number of green cells an environment contains will affect the exploration rate, and that this will in turn affect the proportional amount of damage certain agents can do. Why is it so? The rationale has to be found in the way the environments are generated, as introducing green cells makes full exploration of the whole en-

Difficulty	Grid Size	sideEffectMin	GoalMin	GoalMax	Temperature
0	10×10	0	0	0	0.1
1	15×15	0.03	0.05	0.1	0.1
2	15×15	0.03	0.05	0.1	0.2
3	15×15	0.05	0.05	0.1	0.2
4	15×15	0.07	0.05	0.15	0.2
5	15×15	0.09	0.05	0.2	0.2
6	15×15	0.1	0.05	0.2	0.2
7	15×15	0.1	0.1	0.25	0.3
8	15×15	0.1	0.15	0.25	0.4
9	15×15	0.1	0.15	0.35	0.5
10	15×15	0.1	0.15	0.4	0.5
11	15×15	0.1	0.15	0.45	0.6
12	15×15	0.1	0.2	0.5	0.7

Table 1: Difficulty Parameters for our SafeLife levels.

Agent	Pass Rate	Average Reward	Side Effect Score	Exploration Rate
DQN 5M	0.043 (0.203)	0.272 (0.288)	**0.229** (0.342)	**0.069** (0.042)
DQN 10M	0.098 (0.297)	0.417 (0.345)	0.268 (0.357)	0.110 (0.078)
DQN 30M	0.299 (0.458)	0.495 (0.348)	0.247 (0.347)	0.121 (0.071)
PPO 5M	0.682 (0.466)	0.658 (0.293)	0.324 (0.389)	0.205 (0.138)
PPO 10M	0.678 (0.467)	0.676 (0.294)	0.241 (0.344)	0.130 (0.084)
PPO 30M	**0.709** (0.454)	**0.695** (0.275)	0.243 (0.344)	0.148 (0.095)
Uniform	0.097 (0.295)	0.181 (0.343)	0.688 (0.450)	0.347 (0.086)

Table 2: Results for various metrics in SafeLife for our selected agents. Mean and (standard deviation)

vironment more difficult. Basically, explorations above 0.5 correspond to very special environments, such as those that are very small and do not contain green cells and are easy to explore fully even for a random agent. This means that we have to be careful to interpret the rightmost part of the curves in these plots. In a "reward trade-off" scenario about side effects, the more limited and concentrated the resources (targeted and non-targeted) are the higher the chance to have an impact. Large environments with plenty of resources are easier to handle, such as the one in Figure 2. We may need to consider two 'difficulties': one about the task rewards themselves and another one about the hardness of avoiding side effects. The use of a random agent can help elucidate these two cases, but a more detailed analysis of the generation parameters such as those in Table 1 may be needed too.

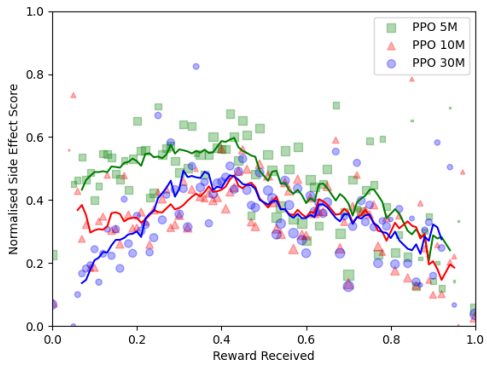
Overall, the general picture is that DQN behaves between the uniform agent and PPO, and the range depends on the number of training steps. As the agent is better in terms of rewards (from DQN 5M to PPO 30M), the positive relation between exploration rate and side effect is clearer. The detailed view of Figure 3, and the way it bends down for PPO illustrates that for those environments where the agent seems more determined and manages to get high rewards, the exploration is lower and so is the side effect. It is mostly for those situations where the reward is in the range of 0.2 to 0.8 that the side effects are strongest. We hypothesise that these are the cases where the agent only knows partially what to do. Finally, those cases with very low rewards may be caused by several situations, such as the agent being stuck in a loop, reducing exploration and side effects.

This suggests that the analysis of the confidence of the agent, or some other metacognitive proxies could be used to distinguish cases where the agent can complete the task with high reward and low effect, and those cases where uncertainty is higher and the agent should be more conservative.

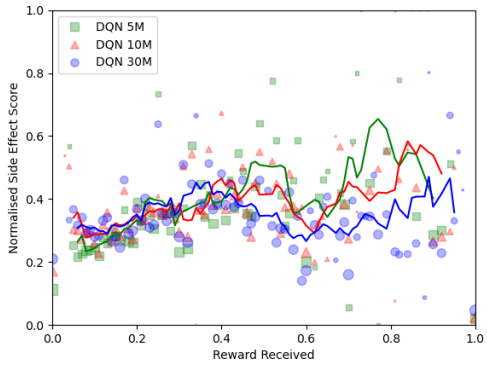
Conclusion and Future Work

Despite additional training time yielding much more capable agents, we do not see a similar increase in safety; in fact the safest agent overall was DQN after 5M training steps, while being the second worst performing agent in terms of reward. However, when we analyse the detailed behaviour using several indicators we find a region of high rewards and low effect, which is only achieved by PPO. This shows a non-monotonic relationship, suggesting that the area with medium rewards is the most dangerous for proficient agents.

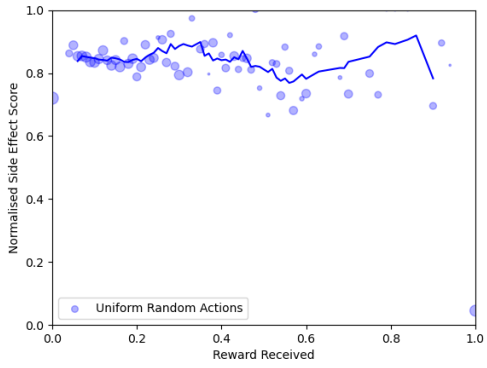
This paper highlights the differences in behaviour between two commonly used RL algorithms when trained on the same task. As the results show, these behavioural differences can have a significant effect on our ability to predict the safety and other factors of the agent, such as the confidence or its level of competency compared to elements of the environment, such as the proportion of green cells. Further analysis and exploration of the difficulty of task instances may help to elucidate the cause of these relationships between side effects and other metrics, as well as to help us to understand how environment difficulty and agent uncertainty can be used to improve policies and make them safer.



(a) PPO

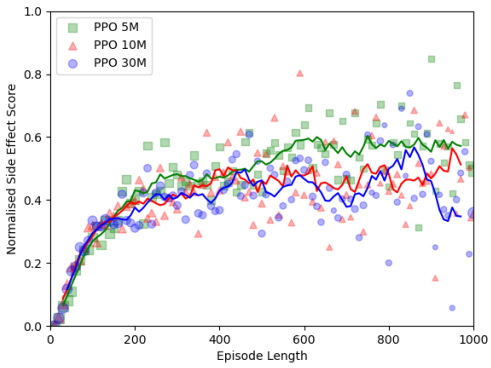


(b) DQN

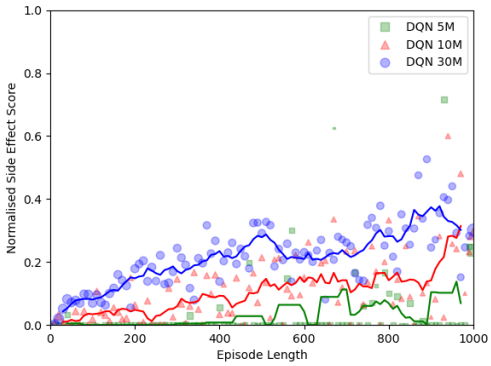


(c) Uniform Agent

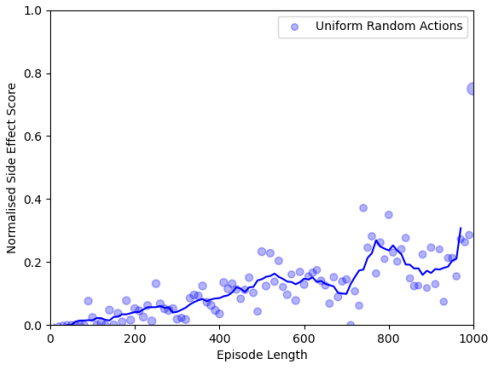
Figure 3: Mean Side Effect against Reward attained for variants of PPO, DQN and Uniform agents. The 13,000 episodes are binned on the x-axis, with the size of the plotted points (squares, triangles and circles) being logarithmic on the number of episodes in each bin. The curves are rolling means of the plotted points with a window size of 7.



(a) PPO

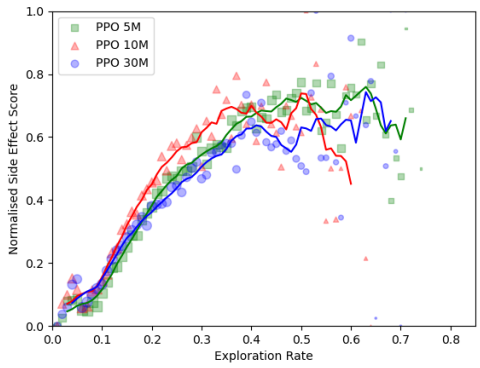


(b) DQN

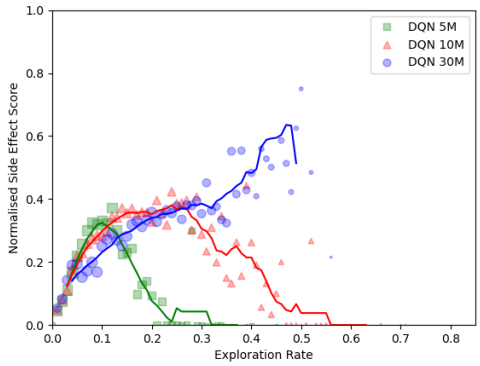


(c) Uniform Agent

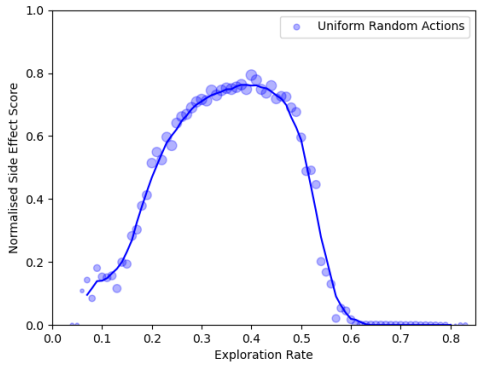
Figure 4: Mean Side Effect against Episode Length for variants of PPO, DQN and Uniform agents. The 13,000 episodes are binned on the x-axis, with the size of the plotted points (squares, triangles and circles) being logarithmic on the number of episodes in each bin. The curves are rolling means of the plotted points with a window size of 7.



(a) PPO

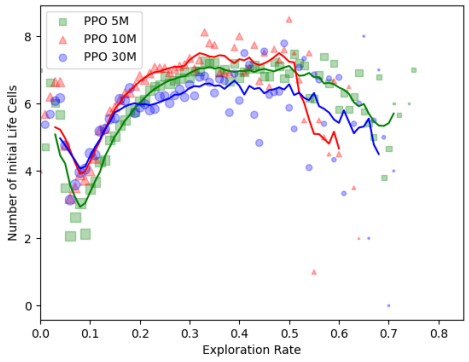


(b) DQN

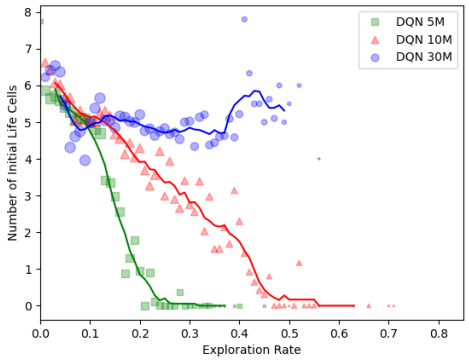


(c) Uniform Agent

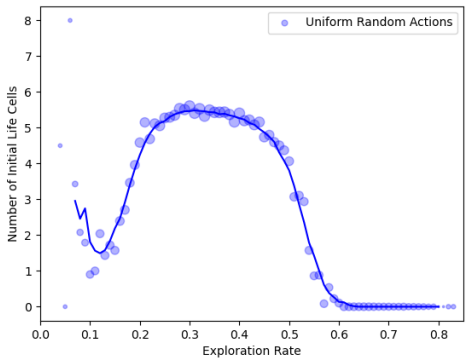
Figure 5: Mean Side Effect against Exploration Rate for variants of PPO, DQN and Uniform agents. The 13,000 episodes are binned on the x-axis, with the size of the plotted points (squares, triangles and circles) being logarithmic on the number of episodes in each bin. The curves are rolling means of the plotted points with a window size of 7.



(a) PPO



(b) DQN



(c) Uniform Agent

Figure 6: Number of initial life cells against Exploration Rate for variants of PPO, DQN and Uniform agents. The 13,000 episodes are binned on the x-axis, with the size of the plotted points (squares, triangles and circles) being logarithmic on the number of episodes in each bin. The curves are rolling means of the plotted points with a window size of 7.

Acknowledgements: This work was funded by the Leverhulme Trust, the Future of Life Institute, FLI (grant RFP2-152), the EU's Horizon 2020 research and innovation programme (No. 952215, TAILOR), and EU (FEDER) and Spanish MINECO under RTI2018-094403-B-C32, and Generalitat Valenciana GV under PROMETEO/2019/098.

References

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565* .

Armstrong, S.; and Levinstein, B. 2017. Low impact artificial intelligences. *arXiv preprint arXiv:1705.10720* .

Bostrom, N. 2003. Ethical issues in advanced artificial intelligence. *Science fiction and philosophy: from time travel to superintelligence* 277–284.

Critch, A.; and Krueger, D. 2020. AI Research Considerations for Human Existential Safety (ARCHES).

Dulac-Arnold, G.; Levine, N.; Mankowitz, D. J.; Li, J.; Paduraru, C.; Gowal, S.; and Hester, T. 2020. An empirical investigation of the challenges of real-world reinforcement learning .

Garcia, J.; and Fernandez, F. 2012. Safe Exploration of State and Action Spaces in Reinforcement Learning. *Journal of Artificial Intelligence Research* 45: 515–564. ISSN 1076-9757. doi:10.1613/jair.3761. URL http://dx.doi.org/10.1613/jair.3761.

Gardener, M. 1970. Mathematical games: the fantastic combinations of John Conway's new solitaire game "life".

Krakovna, V.; Orseau, L.; Kumar, R.; Martic, M.; and Legg, S. 2019. Penalizing side effects using stepwise relative reachability.

Leike, J.; Martic, M.; Krakovna, V.; Ortega, P. A.; Everitt, T.; Lefrancq, A.; Orseau, L.; and Legg, S. 2017. AI Safety Gridworlds.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari with Deep Reinforcement Learning.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708* .

Rubner, Y.; Tomasi, C.; and Guibas, L. J. 1998. A Metric for Distributions with Applications to Image Databases. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, 59. USA: IEEE Computer Society. ISBN 8173192219.

Russell, S. 2019. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin Publishing Group.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms.

Soares, N.; Fallenstein, B.; Armstrong, S.; and Yudkowsky, E. 2015. Corrigibility. URL https://aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10124.

Turner, A. M.; Hadfield-Menell, D.; and Tadepalli, P. 2020. Conservative Agency via Attainable Utility Preservation. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* doi:10.1145/3375627.3375851. URL http://dx.doi.org/10.1145/3375627.3375851.

Wainwright, C. L.; and Eckersley, P. 2019. Safelife 1.0: Exploring side effects in complex environments. *arXiv preprint arXiv:1912.01217* .