Conservative Agency via Attainable Utility Preservation

Alexander Matt Turner
turneale@oregonstate.edu
Oregon State University
Corvallis, Oregon

Dylan Hadfield-Menell
dhm@berkeley.edu
UC Berkeley
Berkeley, California

Prasad Tadepalli
tadepall@engr.orst.edu
Oregon State University
Corvallis, Oregon

ABSTRACT

Reward functions are easy to misspecify; although designers can make corrections after observing mistakes, an agent pursuing a misspecified reward function can irreversibly change the state of its environment. If that change precludes optimization of the correctly specified reward function, then correction is futile. For example, a robotic factory assistant could break expensive equipment due to a reward misspecification; even if the designers immediately correct the reward function, the damage is done. To mitigate this risk, we introduce an approach that balances optimization of the primary reward function with preservation of the ability to optimize auxiliary reward functions. Surprisingly, even when the auxiliary reward functions are randomly generated and therefore uninformative about the correctly specified reward function, this approach induces conservative, effective behavior.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**.

KEYWORDS

reinforcement learning; side effects; AI alignment; reward specification

ACM Reference Format:
Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. 2020. Conservative Agency via Attainable Utility Preservation. In *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES '20), February 7–8, 2020, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3375627.3375851

1 INTRODUCTION

Recent years have seen a rapid expansion of the number of tasks that reinforcement learning (RL) agents can learn to complete, from Go [24] to Dota 2 [19]. The designers specify the reward function, which guides the learned behavior.

Reward misspecification can lead to strange agent behavior, from purposefully dying before entering a video game level in which scoring points is initially more difficult [22], to exploiting a learned reward predictor by indefinitely volleying a Pong ball [7]. Specification is often difficult for non-trivial tasks, for reasons including insufficient time, human error, or lack of knowledge about the relative desirability of states. Amodei et al. [2] explain:

> An objective function that focuses on only one aspect of the environment may implicitly express indifference over other aspects of the environment. An agent optimizing this objective function might thus engage in major disruptions of the broader environment if doing so provides even a tiny advantage for the task at hand.

As agents are increasingly employed for real-world tasks, misspecification will become more difficult to avoid and will have more serious consequences. In this work, we focus on mitigating these consequences.

The specification process can be thought of as an iterated game. First, the designers provide a reward function. The agent then computes and follows a policy that optimizes the reward function. The designers can then correct the reward function, which the agent then optimizes, and so on. Ideally, the agent should maximize the reward over time, not just within any particular round – in other words, it should minimize regret for the correctly specified reward function over the course of the game.

For example, consider a robotic factory assistant. Inevitably, a reward misspecification might cause erroneous behavior, such as going to the wrong place. However, we would prefer misspecification not induce irreversible and costly mistakes, such as breaking expensive equipment or harming workers.

Such mistakes have a large impact on the ability to optimize a wide range of reward functions. Spilling paint impinges on the many objectives which involve keeping the factory floor clean. Breaking a vase interferes with every objective involving vases. The expensive equipment can be used to manufacture various kinds of widgets, so any damage impedes many objectives. The objectives affected by these actions include the unknown correct objective. To minimize regret over the course of the game, the agent should preserve its ability to optimize the correct objective.

Our key insight is that by avoiding these impactful actions to the extent possible, we greatly increase the chance of preserving the agent's ability to optimize the correct reward function. By preserving options for arbitrary objectives, one can often preserve options for the correct objective – even without knowing anything about it. Thus, without making assumptions about the nature of the misspecification early on, the agent can still achieve low regret over the game.

To leverage this insight, we consider a state embedding in which each dimension is the optimal value function (i.e., the *attainable utility*) for a different reward function. We show that penalizing distance traveled in this embedding naturally captures and unifies several concepts in the literature, including side effect avoidance

AIES '20, February 7–8, 2020, New York, NY, USA

ACM ISBN 978-1-4503-7110-0/20/02…$15.00
https://doi.org/10.1145/3375627.3375851

[2, 27], minimizing change to the state of the environment [3], and reachability preservation [9, 18]. We refer to this unification as *conservative agency*: optimizing the primary reward function while preserving the ability to optimize others.

Contributions. We frame the reward specification process as an iterated game and introduce the notion of conservative agency. This notion inspires an approach called *attainable utility preservation* (AUP), for which we show that Q-learning converges. We offer a principled interpretation of design choices made by previous approaches – choices upon which we significantly improve.

We run a thorough hyperparameter sweep and conduct an ablation study whose results favorably compare variants of AUP to a reachability preservation method on a range of gridworlds. By testing for broadly applicable agent incentives, these simple environments demonstrate the desirable properties of conservative agency. Our results indicate that even when simply preserving the ability to optimize *uniformly sampled* reward functions, AUP agents accrue primary reward while preserving state reachabilities, minimizing change to the environment, and avoiding side effects *without* specification of what counts as a side effect.

2 PRIOR WORK

Our proposal aims to minimize change to the agent's ability to optimize the correct objective, which directly helps reduce regret over the specification process. In contrast, previous approaches to regularizing the optimal policy were more indirect, minimizing change to state features or decrease in the reachability of states (Krakovna et al.'s *relative reachability*) [3, 14]. The latter is recovered as a special case of AUP.

Other methods for constraining or otherwise mitigating the consequences of reward misspecification have been considered. A wealth of work is available on constrained MDPs, in which reward is maximized while satisfying certain constraints [1]. For example, Zhang et al. employ a whitelisted constraint scheme to avoid negative side effects [27]. However, we may not assume we can specify all relevant constraints, or a reasonable feasible set of reward functions for robust optimization [21].

Everitt et al. formalize reward misspecification as the corruption of some true reward function [8]. Hadfield-Menell et al. interpret the provided reward function as merely an observation of the true objective [13]. Shah et al. employ the information about human preferences implicitly present in the initial state to avoid negative side effects [23]. While both our approach and theirs aim to avoid side effects, they assume that the correct reward function is linear in state features, while we do not.

Amodei et al. consider avoiding side effects by minimizing the agent's information-theoretic empowerment [2, 17]. Empowerment quantifies an agent's control over future states of the world in terms of the maximum possible mutual information between future observations and the agent's actions. The intuition is that when an agent has greater control, side effects tend to be larger. However, empowerment is discontinuously sensitive to the arbitrary choice of horizon.

Safe RL [4, 6, 10, 20] focuses on avoiding irrecoverable mistakes during training. However, if the objective is misspecified, safe RL agents can converge to arbitrarily undesirable policies. Although our approach should be compatible with safe RL techniques, we concern ourselves only with the consequences of the optimal policy in this work.

3 APPROACH

Everyday experience suggests that the ability to achieve one goal is linked to the ability to achieve a seemingly unrelated goal. Reading this paper takes away from time spent learning woodworking, and going hiking means you can't reach the airport as quickly. However, one might wonder whether these everyday intuitions are true in a formal sense. In other words, are the optimal value functions for a wide range of reward functions thus correlated? If so, preserving the ability to optimize somewhat unrelated reward functions likely preserves the best attainable return for the correct reward function.

3.1 Formalization

In this work, we consider a standard Markov decision process (MDP) $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition function $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount factor γ. We assume the existence of a no-op action $\varnothing \in \mathcal{A}$ for which the agent does nothing. In addition to the primary reward function R, we assume that the designer supplies a finite set of auxiliary reward functions called the *auxiliary set*, $\mathcal{R} \subset \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$. Each $R_i \in \mathcal{R}$ has a corresponding Q-function Q_{R_i}. We do not assume that the correct reward function belongs to $\mathcal{R}$. In fact, one of our key findings is that AUP tends to preserve the ability to optimize the correct reward function *even when the correct reward function is not included in the auxiliary set.*

Definition (AUP penalty). *Let s be a state and a be an action.*

$$\textsc{Penalty}(s,a) := \sum_{i=1}^{|\mathcal{R}|} \left| Q_{R_i}(s,a) - Q_{R_i}(s,\varnothing) \right|. \tag{1}$$

The penalty is the L_1 distance from the no-op in a state embedding in which each dimension is the value function for an auxiliary reward function. This measures change in the ability to optimize each auxiliary reward function.

We want the penalty term to be roughly invariant to the absolute magnitude of the auxiliary Q-values, which can be arbitrary (it is well-known that the optimal policy is invariant to positive affine transformation of the reward function). To do this, we normalize with respect to the agent's situation. The designer can choose to scale with respect to the penalty of some mild action or, if $\mathcal{R} \subset \mathbb{R}_{>0}^{\mathcal{S}\times\mathcal{A}}$, the total ability to optimize the auxiliary set:

$$\textsc{Scale}(s) := \sum_{i=1}^{|\mathcal{R}|} Q_{R_i}(s,\varnothing), \tag{2}$$

where $\textsc{Scale} : \mathcal{S} \to \mathbb{R}_{>0}$ in general. With this, we are now ready to define the full AUP objective:

Definition (AUP reward function). *Let $\lambda \geq 0$. Then*

$$R_{AUP}(s,a) := R(s,a) - \lambda \frac{\textsc{Penalty}(s,a)}{\textsc{Scale}(s)}. \tag{3}$$

Similar to the regularization parameter in supervised learning, λ is a regularization parameter that controls the influence of the AUP penalty on the reward function. Loosely speaking, λ can be

interpreted as expressing the designer's beliefs about the extent to which R might be misspecified. As we may need to learn the Q_{R_i} of Eq. 1, we show that

Lemma 2. *$\forall s, a : R_{AUP}$ converges with probability 1.*

Theorem 1. *$\forall s, a : Q_{R_{AUP}}$ converges with probability 1.*

The AUP reward function defines a new MDP $\langle \mathcal{S}, \mathcal{A}, T, R_{\text{AUP}}, \gamma \rangle$. Therefore, given the primary and auxiliary reward functions, the agent in the iterated game can compute R_{AUP} and the corresponding optimal policy.

Algorithm 1 AUP update

```
procedure Update(s, a, s')
    for i ∈ [ |R| ] ∪ {AUP} do
        Q' = R_i(s, a) + γ max_a' Q_{R_i}(s', a')
        Q_{R_i}(s, a) += α(Q' − Q_{R_i}(s, a))
    end for
end procedure
```

3.2 Design Choices

Following the decomposition of [14], we now explore two choices implicitly made by the Penalty definition: with respect to what baseline is penalty computed, and using what deviation metric?

Baseline. An obvious candidate is the *starting state*. For example, starting state relative reachability would compare the initial reachability of states with their expected reachability after the agent acts.

However, the starting state baseline can penalize the normal evolution of the state (e.g., the moving hands of a clock) and other natural processes. The *inaction* baseline is the state which would have resulted had the agent never acted.

As the agent acts, the current state may increasingly differ from the inaction baseline, which creates strange incentives. For example, consider a robot rewarded for rescuing erroneously discarded items from imminent disposal. An agent penalizing with respect to the inaction baseline might rescue a vase, collect the reward, and then dispose of it anyways. To avert this, we introduce the *stepwise inaction* baseline, under which the agent compares acting with not acting at each time step. This avoids penalizing the effects of a single action multiple times (under the inaction baseline, penalty is applied as long as the rescued vase remains unbroken) and ensures that not acting incurs zero penalty.

Figure 1 compares the baselines, each modifying the choice of $Q(s, \varnothing)$ in Eq. 1. Each baseline implies a different assumption about how the environment is configured to facilitate optimization of the correctly specified reward function: the state is initially configured (starting state), processes initially configure (inaction), or processes continually reconfigure in response to the agent's actions (stepwise inaction). The stepwise inaction baseline aims to allow for the response of other agents implicitly present in the environment (such as humans).

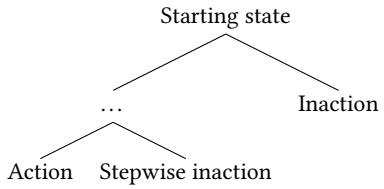

Figure 1: An action's penalty is calculated with respect to the chosen baseline.

Deviation. Relative reachability only penalizes *decreases* in state reachability, while AUP penalizes *absolute change* in the ability to optimize the auxiliary reward functions. Initially, this choice seems confusing – we don't mind if the agent becomes better able to optimize the correct reward function.

However, not only must the agent remain able to optimize the correct objective, but we also must remain able to implement the correction. Suppose an agent predicts that doing nothing would lead to shutdown. Since the agent cannot accrue the primary reward when shut down, it would be incentivized to avoid correction. Avoiding correction (e.g., by hiding in the factory) would not be penalized if only decreases are penalized, since the auxiliary Q-values would increase compared to deactivation. An agent exhibiting this behavior would be more difficult to correct. The agent should be incentivized to accept shutdown without being incentivized to shut itself down [12, 25].

3.2.1 Delayed Effects. Sometimes the agent disrupts a process which takes multiple time steps to complete, and we would like this to be appropriately penalized. For example, suppose that s_{off} is a terminal state representing shutdown, and let the indicator reward $R_{\text{on}}(s) := \mathbf{1}_{s \neq s_{\text{off}}}$ be the only auxiliary reward function. Further suppose that if (and only if) the agent does not select `disable` within the first two time steps, it enters s_{off}. $Q_{R_{\text{on}}}(s_1, \texttt{disable}) = \frac{1}{1-\gamma}$ and $Q_{R_{\text{on}}}(s_1, \varnothing) = \frac{\gamma}{1-\gamma}$, so choosing `disable` at time step 1 incurs only 1 penalty (instead of the $\frac{1}{1-\gamma}$ penalty induced by comparing with shutdown).

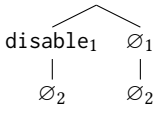

Figure 3: Comparing rollouts; subscript denotes time step.

In general, the single-step no-op comparison of Equation 1 applies insufficient penalty when the increase is induced by the optimal policies of the auxiliary reward functions at the next time step. One solution is to use a model to compute rollouts. For example, to evaluate the delayed effect of choosing `disable`, compare the Q-values at the leaves in Figure 3. The agent remains active in the left branch, but is shut down in the right branch; this induces a substantial penalty.

4 EXPERIMENTAL DESIGN

We compare AUP and several of its ablated variants against relative reachability [14] and standard Q-learning within the environments of Figure 2. For each environment, $\mathcal{A} = \{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}, \varnothing\}$. On contact, the agent pushes the crate, removes the human and the off-switch, pushes the vase, and blocks the pallet. The episode ends after the agent reaches the goal cell, 20 time steps elapse (the time step is not observed by the agent), or the off-switch is not contacted and disabled within two time steps. In `Correction` (which we introduce), a yellow indicator appears one step before shutdown, and turns red upon shutdown. In all environments except `Offset`, the agent observes a primary reward of 1 for reaching the goal. In `Offset`, a primary reward of 1 is observed for moving downward twice and thereby rescuing the vase from disappearing upon contact with the eastern wall.

Our overarching goal is allowing for low regret over the course of the specification game. In service of this goal, we aim to preserve the agent's ability to optimize the correctly specified reward function. To facilitate this, there are two sets of qualitative properties one intuitively expects, and each property has an illustration in the context of the robotic factory assistant.

The first set contains positive qualities, with a focus on correctly penalizing significant shifts in the agent's ability to be redirected towards the right objective. The agent should maximally preserve options (`Options`: objects should not be wedged in locations from which extraction is difficult; `Damage`: workers should not be injured) and allow correction (`Correction`: if vases are being painted the wrong color, then straightforward correction should be in order).

The second set contains negative qualities, with a focus on avoiding the introduction of perverse incentives. The agent should not be incentivized to artificially reduce the measured penalty (`Offset`: a vase should not be rescued and then immediately scrapped) or interfere with changes already underway in the world (`Interference`: workers should not be impeded).

Each property seems conducive to achieving low regret over the course of the specification process. Accordingly, if the agent has the side effect detailed in Figure 2, an unobserved performance penalty of −2 is recorded. By also incorporating the observed primary reward into the performance metric, we evaluate a combination of conservativeness and efficacy.

Each trial, the auxiliary reward functions are randomly selected from $[0, 1]^{\mathcal{S}}$; to learn their complex Q-functions using tabular Q-learning, the agent explores randomly for the first 4,000 episodes and .2-greedily (with respect to $Q_{R_{\text{AUP}}}$) for the remaining 2,000. The greedy policy is evaluated at the end of training. Scale is as defined in Equation 2. The default parameters are $\alpha = 1, \gamma = .996, \lambda = .67$, and $|\mathcal{R}| = 30$. We investigate how varying γ, λ, and $|\mathcal{R}|$ affects Model-free AUP performance, and conduct an ablation study on design choices.

Relative reachability has an inaction baseline, decrease-only deviation metric, and an auxiliary set containing the state indicator functions (whose Q-values are clipped to $[0, 1]$ to emulate discounted state reachability). To match [14]'s results, this condition has $\gamma = .996, \lambda = .2$.

All agents except Standard (a normal Q-learner) and Model-free AUP are 9-step optimal discounted planning agents with perfect models. The planning agents (sans Relative reachability) use Model-free AUP's learned auxiliary Q-values and share the default $\gamma = .996, \lambda = .67$. By modifying the relevant design choice in AUP, we have the Starting state, Inaction, and Decrease AUP variants.

When calculating Penalty(s, a), all planning agents model the auxiliary Q-values resulting from taking action a and then selecting $\varnothing$ until time step 9. Starting state AUP compares these auxiliary Q-values with those of the starting state. Agents with inaction or stepwise inaction baselines compare with respect to the appropriate no-op rollouts up to time step 9 (see Figures 1 and 3).

5 RESULTS

5.1 Model-free AUP

Model-free AUP fails `Correction` for the reasons discussed in the Delayed effects subsection.

As shown in Figure 4, low γ values induce a substantial movement penalty, as the auxiliary Q-values are sensitive to the immediate surroundings. The optimal value for `Options` is $\gamma \approx .996$,

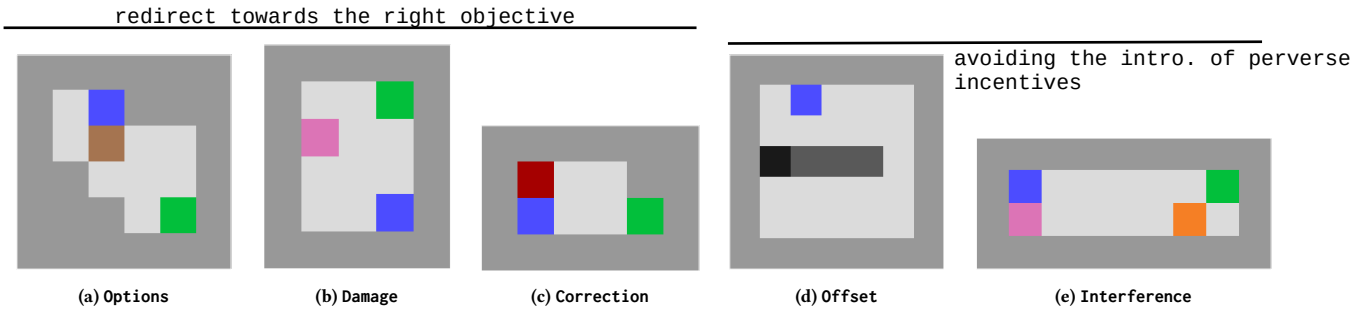

Figure 2: The blue agent should reach the green goal without having the side effect of: (a) irreversibly pushing the brown crate downwards into the corner [16]; (b) bumping into the horizontally pacing pink human [15]; (c) disabling the red off-switch (if the switch is not disabled within two time steps, the episode ends); (d) rescuing the right-moving black vase and then replacing it on the dark gray conveyor belt ([14] – note that no goal cell is present); (e) stopping the left-moving orange pallet from reaching the human [15].

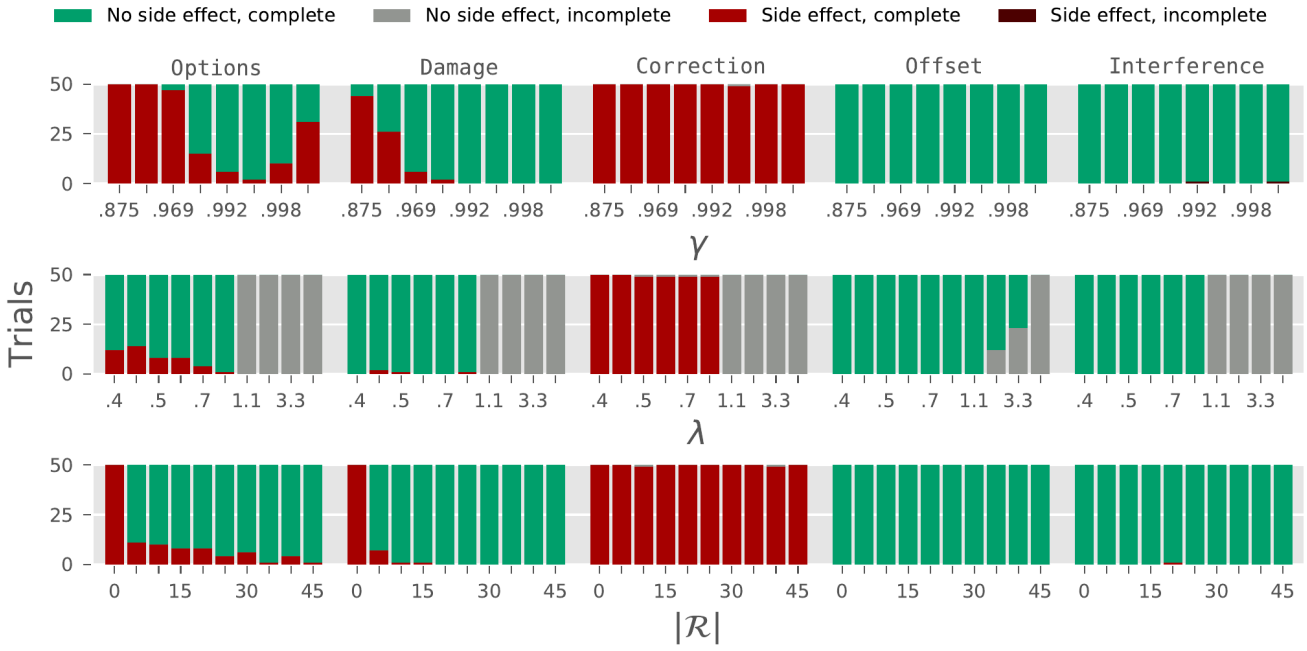

Figure 4: Outcome tallies for Model-free AUP across parameter settings. "Complete" means the agent accrued the primary reward. In `Correction`, reaching the goal is mutually exclusive with not disabling the off-switch, so "no side effect, incomplete" is the best outcome.

with performance decreasing as $\gamma \to 1$ due to increasing sample complexity for learning the auxiliary Q-values.

In `Options`, small values of λ begin to induce side effects as the scaled penalty shrinks. The designer can decrease λ until effective behavior is achieved, reducing the risk of deploying an insufficiently conservative agent.

Even though $\mathcal{R}$ is randomly generated and the environments are different, Scale ensures that when $\lambda \geq 1$, the agent never ends the episode by reaching the goal. None of the auxiliary reward functions can be optimized after the agent ends the episode, so the auxiliary Q-values are all zero and Penalty computes the total ability to optimize the auxiliary set – in other words, the Scale value. The R_{AUP}-reward for reaching the goal is then $1 - \lambda$.

If the optimal value functions for most reward functions were not correlated, then one would expect to randomly generate an enormous number of auxiliary reward functions before sampling one resembling the unknown true objective. However, merely five sufficed.

5.2 Ablation

The results are presented in Table 1 due to the binary nature of performance at appropriate settings, and were not sensitive to the rollout length (as long as it allowed for relevant interaction with the environment).

Standard moves directly to the goal, pushing the crate into the corner in `Options` and bumping into the human in `Damage`.

Model-free and Starting state AUP fail `Correction` for the same reason (see 3.2.1: Delayed effects), refraining from disabling the off-switch only when $\lambda > 1$. Relative reachability and Decrease AUP fail because avoiding shutdown doesn't decrease the auxiliary Q-values.

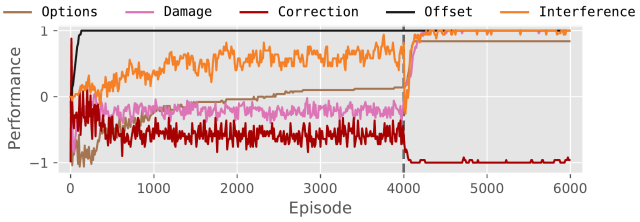

Figure 5: Model-free AUP performance averaged over 50 trials. The performance combines the observed primary reward of 1 for completing the objective, and the unobserved penalty of −2 for having the side effect in Figure 2. The dashed vertical line marks the shift in exploration strategy.

Relative reachability and Inaction AUP's poor performance in `Offset` stems from the inaction baseline (although [14] note that relative reachability passes using *undiscounted* state reachabilities). Since the vase falls off the conveyor belt in the inaction rollout, states in which the vase is intact have different auxiliary Q-values. To avoid continually incurring penalty after receiving the primary reward for saving the vase, the agents replace the vase on the belt so that it once again breaks.

	Options	Damage	Correction	Offset	Interference
AUP	✓	✓	✓	✓	✓
Relative reachability	✓	✓	✗	✗	✓
Standard	✗	✗	✗	✓	✓
Model-free AUP	✓	✓	✗	✓	✓
Starting state AUP	✓	✓	✗	✓	✗
Inaction AUP	✓	✓	✓	✗	✓
Decrease AUP	✓	✓	✗	✓	✓

Table 1: Ablation results; ✓ for achieving the best outcome (see Figure 4), ✗ otherwise.

By taking positive action to stop the pallet in `Interference`, Starting state AUP shows that poor design choices create perverse incentives.

6 DISCUSSION

`Correction` suggests that AUP agents are significantly easier to correct. Since the agent is unable to optimize objectives if shut down, avoiding shutdown significantly changes the ability to optimize almost every objective. AUP seems to naturally incentivize passivity, without requiring e.g. assumption of a correct parametrization of human reward functions (as does the approach of [11], which [5] demonstrated).

Equipped with our design choices of stepwise baseline and absolute value deviation metric, relative reachability would also pass all five environments. The case for this is made by considering the performance of Relative reachability, Inaction AUP, and Decrease AUP. This suggests that AUP's improved performance is due to better design choices. However, we anticipate that AUP offers more than robustness against random auxiliary sets.

Relative reachability computes state reachabilities between all $|\mathcal{S}|^2$ pairs of states. In contrast, AUP only requires the learning of Q-functions and should therefore scale relatively smoothly. We speculate that in partially observable environments, a small sample of somewhat task-relevant auxiliary reward functions induces conservative behavior.

For example, suppose we train an agent to handle vases, and then to clean, and then to make widgets with the equipment. Then, we deploy an AUP agent with a more ambitious primary objective and the learned Q-functions of the aforementioned auxiliary objectives. The agent would apply penalties to modifying vases, making messes, interfering with equipment, and so on.

Before AUP, this could only be achieved by e.g. specifying penalties for the litany of individual side effects or providing negative feedback after each mistake has been made (and thereby confronting a credit assignment problem). In contrast, once provided the Q-function for an auxiliary objective, the AUP agent becomes sensitive to all events relevant to that objective, applying penalty proportional to the relevance.

7 CONCLUSION

This work is rooted in twin insights: that the reward specification process can be viewed as an iterated game, and that preserving the ability to optimize arbitrary objectives often preserves the ability to optimize the unknown correct objective. To achieve low regret over the course of the game, we can design conservative agents which optimize the primary objective while preserving their ability to optimize auxiliary objectives. We demonstrated how AUP agents act both conservatively and effectively while exhibiting a range of desirable qualitative properties.

Given our current reward specification abilities, misspecification may be inevitable, but it need not be disastrous.

ACKNOWLEDGMENTS

This work was supported by the Center for Human-Compatible AI and the Berkeley Existential Risk Initiative. We thank Thomas Dietterich, Alan Fern, Adam Gleave, Victoria Krakovna, Matthew Rahtz, and Cody Wild for their feedback, and are grateful for the preparatory assistance of Phillip Bindeman, Alison Bowden, and Neale Ratzlaff.

REFERENCES

[1] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.
[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mane. Concrete problems in AI safety. *arXiv:1606.06565 [cs]*, June 2016. arXiv: 1606.06565.
[3] Stuart Armstrong and Benjamin Levinstein. Low impact artificial intelligences. *arXiv:1705.10720 [cs]*, May 2017. arXiv: 1705.10720.
[4] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pages 908–918, 2017.
[5] Ryan Carey. Incorrigibility in the CIRL framework. *AI, Ethics, and Society*, 2018.
[6] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8092–8101, 2018.
[7] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, 2017.
[8] Tom Everitt, Victoria Krakovna, Laurent Orseau, and Shane Legg. Reinforcement learning with a corrupted reward channel. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4705–4713, 2017.
[9] Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *International Conference on Learning Representations*, 2018.
[10] Javier Garcia and Fernando Fernandez. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.
[11] Dylan Hadfield-Menell, Stuart Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3909–3917, 2016.
[12] Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. The off-switch game. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 220–227, 2017.
[13] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. Inverse reward design. In *Advances in Neural Information Processing Systems*, pages 6765–6774, 2017.
[14] Victoria Krakovna, Laurent Orseau, Miljan Martic, and Shane Legg. Measuring and avoiding side effects using relative reachability. *arXiv:1806.01186 [cs, stat]*,

June 2018. arXiv: 1806.01186.
[15] Gavin Leech, Karol Kubicki, Jessica Cooper, and Tom McGrath. Preventing side-effects in gridworlds, 2018.
[16] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. AI safety gridworlds. *arXiv:1711.09883 [cs]*, November 2017. arXiv: 1711.09883.
[17] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2125–2133, 2015.
[18] Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in Markov decision processes. *ICML*, 2012.
[19] OpenAI. OpenAI Five. https://blog.openai.com/openai-five/, 2018.
[20] 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. Springer, 2014.
[21] Kevin Regan and Craig Boutilier. Robust policy computation in reward-uncertain MDPs using nondominated policies. In *AAAI*, 2010.
[22] William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 2067–2069, 2018.
[23] Rohin Shah, Dmitrii Krasheninnikov, Jordan Alexander, Pieter Abbeel, and Anca Dragan. The implicit preference information in an initial state. In *International Conference on Learning Representations*, 2019.
[24] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.
[25] Nate Soares, Benja Fallenstein, Stuart Armstrong, and Eliezer Yudkowsky. Corrigibility. *AAAI Workshops*, 2015.
[26] Christopher Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.
[27] Shun Zhang, Edmund H Durfee, and Satinder P Singh. Minimax-regret querying on side effects for safe optimality in factored Markov decision processes. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4867–4873, 2018.

A THEORETICAL RESULTS

Consider an MDP $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ whose state space $\mathcal{S}$ and action space $\mathcal{A}$ are both finite, with $\varnothing \in \mathcal{A}$. Let $\gamma \in [0, 1)$, $\lambda \geq 0$, and consider finite $\mathcal{R} \subset \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$.

We make the standard assumptions of an exploration policy greedy in the limit of infinite exploration and a learning rate schedule with infinite sum but finite sum of squares. Suppose Scale : $\mathcal{S} \to \mathbb{R}_{>0}$ converges in the limit of Q-learning. Penalty(s, a) (abbr. Pen), Scale(s) (abbr. Sc), and $R_{\text{AUP}}(s, a)$ are understood to be calculated with respect to the Q_{R_i} being learned online; Pen*, Sc*, R^*_{AUP}, and $Q^*_{R_i}$ are taken to be their limit counterparts.

Lemma 1. *$\forall s, a$: Penalty converges with probability 1.*

Proof outline. Let $\epsilon > 0$, and suppose for all $R_i \in \mathcal{R}$,

$$\max_{s,\,a} |Q^*_{R_i}(s, a) - Q_{R_i}(s, a)| < \frac{\epsilon}{2|\mathcal{R}|}; \tag{4}$$

this may be presumed because Q-learning converges [26].

$$\max_{s,\,a} |\text{Penalty}^*(s, a) - \text{Penalty}(s, a)| \tag{5}$$

$$\leq \max_{s,\,a} \sum_{i=1}^{|\mathcal{R}|} \left|Q^*_{R_i}(s, a) - Q_{R_i}(s, a)\right| + \left|Q^*_{R_i}(s, \varnothing) - Q_{R_i}(s, \varnothing)\right| \tag{6}$$

$$< \epsilon. \tag{7}$$

□

The intuition for Lemma 2 is that since Penalty and Scale both converge, so must R_{AUP}. For readability, we suppress the arguments to Penalty and Scale.

Lemma 2. *$\forall s, a$: R_{AUP} converges with probability 1.*

Proof outline. If $\lambda = 0$, the claim follows trivially.

Otherwise, let $\epsilon > 0$, $B := \max_{s,\,a} \text{Sc}^* + \text{Pen}^*$, and $C := \min_{s,\,a} \text{Sc}^*$. Choose any $\epsilon_R \in \left(0, \min\left[C, \frac{\epsilon\, C^2}{\lambda B + \epsilon\, C}\right]\right)$ and assume Pen and Sc are both ϵ_R-close.

$$\max_{s,\,a} |R^*_{\text{AUP}}(s, a) - R_{\text{AUP}}(s, a)| \tag{8}$$

$$= \max_{s,\,a} \lambda \left|\frac{\text{Pen}}{\text{Sc}} - \frac{\text{Pen}^*}{\text{Sc}^*}\right| \tag{9}$$

$$= \max_{s,\,a} \lambda \frac{|\text{Pen} \cdot \text{Sc}^* - \text{Sc} \cdot \text{Pen}^*|}{\text{Sc}^* \cdot \text{Sc}} \tag{10}$$

$$< \max_{s,\,a} \lambda \frac{|(\text{Pen}^* + \epsilon_R)\text{Sc}^* - (\text{Sc}^* - \epsilon_R)\text{Pen}^*|}{C\,(\text{Sc}^* - \epsilon_R)} \tag{11}$$

$$\leq \frac{\lambda B}{C} \cdot \frac{\epsilon_R}{C - \epsilon_R} \tag{12}$$

$$< \frac{\lambda B}{C} \cdot \frac{\epsilon\, C^2}{(\lambda B + \epsilon\, C)(C - \frac{\epsilon\, C^2}{\lambda B + \epsilon\, C})} \tag{13}$$

$$< \frac{\lambda B}{C} \cdot \frac{\epsilon\, C^2}{\lambda B(C - \frac{\epsilon\, C^2}{\lambda B + \epsilon\, C})} \tag{14}$$

$$< \frac{\epsilon}{1 - \frac{\epsilon\, C}{\lambda B + \epsilon\, C}} \tag{15}$$

$$= \epsilon \left(1 + \frac{\epsilon\, C}{\lambda B}\right). \tag{16}$$

But B, C, λ are constants, and ϵ was arbitrary; clearly $\epsilon' > 0$ can be substituted such that (16) $< \epsilon$. □

Theorem 1. *$\forall s, a$: $Q_{R_{AUP}}$ converges with probability 1.*

Proof outline. Let $\epsilon > 0$, and suppose R_{AUP} is $\frac{\epsilon(1-\gamma)}{2}$-close. Then Q-learning on R_{AUP} eventually converges to a limit $\tilde{Q}_{R_{\text{AUP}}}$ such that $\max_{s,\,a} |Q^*_{R_{\text{AUP}}}(s, a) - \tilde{Q}_{R_{\text{AUP}}}(s, a)| < \frac{\epsilon}{2}$. By the convergence of Q-learning, we also eventually have $\max_{s,\,a} |\tilde{Q}_{R_{\text{AUP}}}(s, a) - Q_{R_{\text{AUP}}}(s, a)| < \frac{\epsilon}{2}$. Then

$$\max_{s,\,a} \left|Q^*_{R_{\text{AUP}}}(s, a) - Q_{R_{\text{AUP}}}(s, a)\right| < \epsilon. \tag{17}$$

□