Evaluating the Robustness of Collaborative Agents

Extended Abstract

Paul Knott
University of Nottingham
paulalexknott@gmail.com

Micah Carroll
UC Berkeley
mdc@berkeley.edu

Sam Devlin
Microsoft Research

Kamil Ciosek
Microsoft Research

Katja Hofmann
Microsoft Research

Anca Dragan
UC Berkeley

Rohin Shah
UC Berkeley

ABSTRACT

Artificial agents trained by deep reinforcement learning will likely encounter novel situations after deployment that were never seen during training. Our agent must be *robust* to handle such situations well. However, if we cannot rely on the average training or validation reward as a metric, then how can we effectively evaluate robustness? We take inspiration from the practice of *unit testing* in software engineering. Specifically, we suggest that when designing AI agents that collaborate with humans, designers should search for potential edge cases in *possible partner behavior* and *possible states encountered*, and write tests which check that the behavior of the agent in these edge cases is reasonable. We apply this methodology to build a suite of unit tests for the Overcooked-AI environment, and use this test suite to evaluate three proposals for improving robustness. We find that the test suite provides significant insight into the effects of these proposals that were generally not revealed by looking solely at the average validation reward. **For our full paper, see arxiv.org/abs/2101.05507**.

KEYWORDS

Human-AI collaboration; Robustness; Multi-agent RL.

ACM Reference Format:
Paul Knott, Micah Carroll, Sam Devlin, Kamil Ciosek, Katja Hofmann, Anca Dragan, and Rohin Shah. 2021. Evaluating the Robustness of Collaborative Agents: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

We would like to build agents that *collaborate* with humans in order to help them accomplish their goals, a setting that has recently been tackled with deep RL [2, 6, 8]. While deep RL has shown remarkable success in training agents that perform well on average [1, 11, 13], the learned policies are often not robust to new situations [5], which precludes deployment in many practical settings with stringent robustness requirements [3, 9].

While there are several approaches we could use to improve robustness, it is hard to *evaluate* these approaches. We care about the average reward on the *deployment* distribution, but the true deployment distribution is never available, because the deployment of the agent itself changes the distribution of inputs it receives [10]. Even pairing the learned agent with people in a user study is not usually representative of the performance at deployment time, since in many realistic domains there is a long tail of unusual edge cases that would not be seen during evaluation but would eventually happen after deployment.

Taking inspiration from software engineering, our core insight is that a suite of *unit tests* can provide significant additional robustness information beyond existing metrics such as the average training or validation reward. Our main contribution is a methodology for designing such a unit test suite in the human-AI collaboration paradigm. We demonstrate the utility of our methodology in Overcooked, a two-player cooperative cooking environment [2, 4].

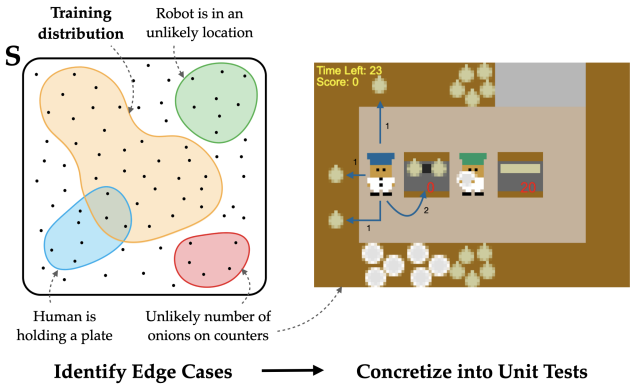

Figure 1: By observing the training distribution and probing the state space, the system designer can identify likely areas that lack robustness, and create unit tests for those areas. On the right is an example of a *state robustness* test. Since the partner (green) is holding a plate, the agent (blue) should get an onion for the left pot, regardless of how its partner plays.

2 DEVELOPING UNIT TESTS

A collaboration task is defined by an environment and a particular partner. So, an agent should be robust to edge cases in the environment (*state robustness*), as well as to the possible partners that

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

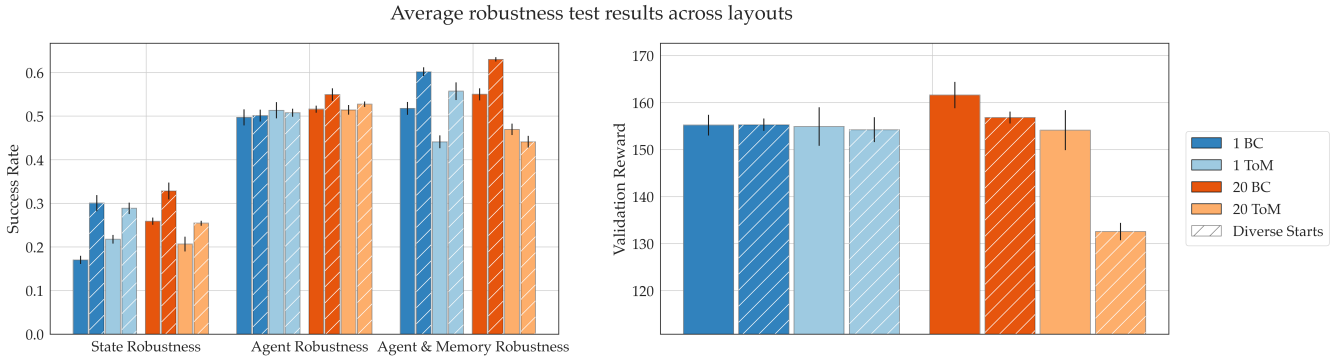

Figure 2: Comparison of unit test scores (LHS) and validation reward (RHS) for deep RL agents trained with and without diverse starts, with and without a population (comprised of 20 human models), and using ToM vs. BC agents.

it must play with (*agent robustness*). Robustness to partners often requires the agent to keep track of history; we term unit tests for this kind of behavior *agent robustness with memory* tests. Given this categorization, our methodology consists of the following steps (for further details, see the full paper [7]): 1) Identify qualitative situations for each test category; 2) Concretize each situation to a unit test; 3) Observe and probe the trained agents to find new situations for which unit tests should be made.

3 EXPERIMENTS

We applied our methodology to Overcooked; an example unit test can be found in Figure 1. The primary goal of our experiments was to evaluate how useful the test suite is in surfacing information about trained agents.

All of the agents we evaluate are trained via deep RL. The starting point we consider for improving robustness is to pair the deep RL agent, during training, with a human model, where as a baseline we use a model trained by behavior cloning (BC) on human-human gameplay data [2]. In our experiments, we manipulate several different factors for further increasing robustness: improving human model quality with a parameterized Theory of Mind (ToM) agent; training with a diverse population of both BC and ToM agents; and initializing from states visited in human-human gameplay ("diverse starts"). Here we only present the results of the *diverse starts* experiments; see the full paper for the remaining results [7].

We report both the average score on unit tests, as well as the validation reward for each agent, computed as the average reward the agent obtains when partnered with human models from a suite of 20 validation agents. This validation reward is not meant as a measure of the robustness of the agents to novel situations, but rather as a baseline to compare the unit test suite against.

Our main hypothesis (**H1**) was that the unit tests and the validation reward would supply different information, and will therefore often not be in agreement. We also expected that using diverse starts would increase state robustness (**H2**).

3.1 Results

Figure 2 shows that, for diverse starts, the unit tests and the validation reward suggest *opposite* conclusions, in agreement with **H1**: we see a notable increase in robustness when using diverse starts, for both state robustness and agent robustness with memory tests, whereas the validation reward either stays the same (for population of 1) or decreases (for population of 20). It appears that using diverse starts confers robustness at the cost of validation reward, an effect that has also been observed with adversarial examples in image classification [12]. On the axis of robustness, we see that **H2** is supported since diverse starts produces an increase in unit test performance across all but one test category and model type.

Hypothesis **H1** was also supported beyond these *diverse starts* results. For example, when using a single ToM agent as the partner, we found that different categories of unit test were affected while average reward stayed the same; and when using a mixed population of BC and ToM agents the unit test robustness remained the same while average reward was improved. Another key finding was that training with a population of 10 BC and 10 ToM agents achieved the greatest robustness.

4 CONCLUSION

In this work, we proposed a methodology for creating a unit test suite to evaluate the robustness of collaborative agents, and showed that such a test suite can provide significant information about robustness that may not be available from just the validation reward.

A given test suite will never be final: there will likely always be more edge cases to include. As different types of failure modes are found or imagined, they can be added into the test suite. We do not claim that unit testing allows us to achieve perfect robustness: rather, we see them as a major improvement over the current status quo of evaluating reward on random rollouts (which only tests the edge cases that are encountered randomly). Current deep RL agents are clearly not robust – none of the agents we tested scored above 65% in Overcooked – suggesting that our approach can serve as a useful metric for the foreseeable future.

ACKNOWLEDGMENTS

This work was partially supported by Open Philanthropy, Microsoft and NSF CAREER. PK acknowledges support from the Royal Commission for the Exhibition of 1851 and from FQXi. We thank researchers at the Center for Human-Compatible AI and the InterACT lab for helpful discussion and feedback.

REFERENCES

[1] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv preprint arXiv:1912.06680* (2019).

[2] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the Utility of Learning about Humans for Human-AI Coordination. In *Advances in Neural Information Processing Systems*. 5175–5186.

[3] A Gasparik, C Gamble, and J Gao. 2018. Safety-first ai for autonomous data centre cooling and industrial control. *DeepMind Blog* (2018).

[4] Ghost Town Games. 2016. Overcooked. https://store.steampowered.com/app/448510/Overcooked/.

[5] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. 2019. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615* (2019).

[6] Hu, Hengyuan and Lerer, Adam and Peysakhovich, Alex and Foerster, Jakob. 2020. "Other-Play" for Zero-Shot Coordination. *arXiv preprint arXiv:2003.02979* (2020).

[7] Paul Knott, Micah Carroll, Sam Devlin, Kamil Ciosek, Katja Hofmann, A. D. Dragan, and Rohin Shah. 2021. Evaluating the Robustness of Collaborative Agents. arXiv:2101.05507 [cs.LG]

[8] Adam Lerer and Alexander Peysakhovich. 2019. Learning Existing Social Conventions via Observationally Augmented Self-Play. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 107–114.

[9] Andrew J Lohn. 2020. Estimating the Brittleness of AI: Safety Integrity Levels and the Need for Testing Out-Of-Distribution Performance. *arXiv preprint arXiv:2009.00802* (2020).

[10] Roland W. Scholz and Claudia R. Binder. 2003. The paradigm of human-environment systems. (2003). https://doi.org/10.3929/ETHZ-A-004520890

[11] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354–359.

[12] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. 2018. Is Robustness the Cost of Accuracy?–A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 631–648.

[13] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/.