Characterising the Robustness of Reinforcement Learning for Continuous Control using Disturbance Injection

Catherine R. Glossop[*]
Department of Engineering Science
University of Toronto
catherine.glossop@robotics.utias.utoronto.ca

Jacopo Panerati
Institute for Aerospace Studies
University of Toronto
jacopo.panerati@utoronto.ca

Amrit Krishnan
Vector Institute
amritk@vectorinstitute.ai

Zhaocong Yuan
Institute for Aerospace Studies
University of Toronto
justin.yuan@mail.utoronto.ca

Angela P. Schoellig
Institute for Aerospace Studies
University of Toronto
angela.schoellig@utoronto.ca

Abstract

In this study, we leverage the deliberate and systematic fault-injection capabilities of an open-source benchmark suite to perform a series of experiments on state-of-the-art deep and robust reinforcement learning algorithms. We aim to benchmark robustness in the context of continuous action spaces—crucial for deployment in robot control. We find that robustness is more prominent for action disturbances than it is for disturbances to observations and dynamics. We also observe that state-of-the-art approaches that are not explicitly designed to improve robustness perform at a level comparable to that achieved by those that are. Our study and results are intended to provide insight into the current state of safe and robust reinforcement learning and a foundation for the advancement of the field, in particular, for deployment in robotic systems.

1 Introduction

Reinforcement learning (RL) has become a promising approach for robotic control, showing how robotic agents can learn to perform a variety of tasks, such as trajectory tracking and goal-reaching, on several robotic systems, from robotic manipulators to self-driving vehicles [9, 21, 24]. While many of these results have been achieved in highly controlled simulated environments [12], the next wave of artificial intelligence (AI) research is now faced with the challenge to deploy in the real world.

When using reinforcement learning to solve real-world problems, safety must be paramount [25, 4, 2, 27, 11, 3]. Unsafe interaction with the environment and/or people in that environment can have very serious consequences, ranging from the destruction of the robot itself to, most importantly, harm to humans. For safety to be guaranteed, an embodied RL agent (i.e., the robot) must be robust to variations in the environment, its dynamics, and unseen situations that can emerge in the real world.

[*]Work done during an internship at the Vector Institute for Artificial Intelligence
An extended version of this work is available at https://arxiv.org/abs/2210.15199.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

In this paper, we quantitatively study and report on the performance of a set of state-of-the-art reinforcement learning approaches in the context of continuous control. We systematically evaluate RL agents (or "controllers") on their performance (i.e., the ability to accomplish the task specified by the environment's reward signal) as well as their robustness [30, 6, 13, 15, 17]. To do so, we use an open-source RL safety benchmarking suite [29]. First, we empirically compare the control policies produced by both traditional and robust RL agents at baseline and then when a variety of disturbances are injected into the environment.

We observe that both the traditional and robust RL agents are more robust to disturbances injected through the actions of the agent, while disturbances injected in the observations and dynamics can cause much more rapid destabilisation. We also note that "vanilla" agents show similar performance to the robust RL agents even when disturbances are injected, despite not being explicitly designed with this purpose in mind. By leveraging open-source simulation, we hope that this work and our insights can provide a basis for further research into safe and robust RL, especially for robot control.

2 Evaluation Setup

Our objective then is to train vanilla and robust RL agents (see additional detail in the Supplementary Material) to perform a task (cart-pole stabilisation) in ideal conditions (i.e., without disturbances) and then assess the robustness of the resulting policies in environments that include injected disturbances in actions, observations, or dynamics (see additional detail in the Supplementary Material).

Each RL agent was trained by randomising the initial state across episodes to improve performance [29] while at test/evaluation time, a unique initial state was used for fairness and consistency. The range of disturbances used in each experiment was selected to include (low) values, at which all or most agents still succeeded in completing the tasks, up until (high) values at which the robustness of all agents eventually fails. In the case of the cart-pole, the goal of the controller is to stabilise the system at a pose of 0 m, or centre, in x and 0 rads in θ, when the pole is upright.

Evaluation Metrics To measure the performance of the control policies, the exponentiated negated quadratic return is averaged over the length of each episode, over 25 evaluation episodes. The same metric was used for training and evaluation.

$$\text{Cost} : J_i^Q = (x_i - x_i^{goal})^T W_x (x_i - x_i^{goal}) + (u_i - u_i^{goal})^T W_u (u_i - u_i^{goal}) \tag{1}$$

$$\text{Ep. Return} : J^R = \sum_{i=0}^{L} \exp(-J_i^Q) \tag{2}$$

$$\text{Avg. Norm. Return} : J_{eval}^R = \frac{1}{N} \sum_{j=0}^{N} \frac{J_j^R}{L_j} \tag{3}$$

Equation (1) shows the task's cost computed at each episode's step i, where x and x^{goal} are the actual and goal states of the system, u and u^{goal} the actual and goal inputs, and W_x and W_u are constant weight matrices. L is the total number of steps in a given episode. Equation (2) shows how to compute the return of an episode j of length L_j from the cost function. L_j is equal (or lower) than the maximum episode duration of 250 steps. Equation (3) shows the average return for N (25) evaluation runs normalised by the length of the run.

3 Results

In the Supplementary Material (Fig. 5), we report the training results when no additional disturbances are applied, showing the reference performance of each controller at baseline. The three algorithms which reach convergence fastest were SAC, PPO, and RAP. SAC and PPO benefit from the stochastic characteristics of their updates. RARL trains more slowly which is what we expect as RARL is also learning to counteract the adversary. However, the same behaviour is not observed for the other robust approach RAP, which also converges quickly, suggesting RAP can be trained more efficiently.

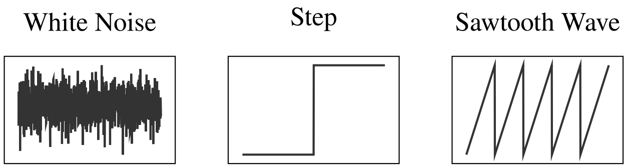

Figure 1: Disturbances injected in the experiments in Section 3: white noise, step, and sawtooth wave.

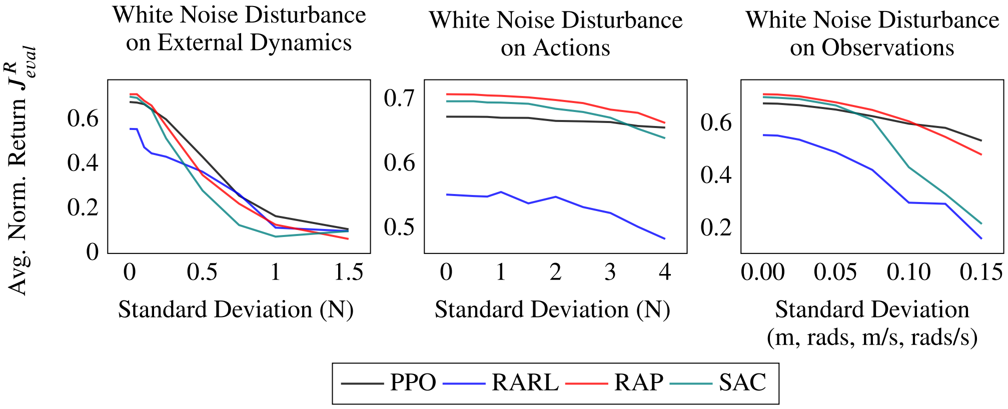

Figure 2: Average normalised return with the injection of white noise disturbances applied to (left to right) dynamics, actions, and observations on the cart-pole stabilisation task for the four RL agents.

3.1 Non-periodic Disturbances

We want to assess the robustness of the trained policies for two vanilla (PPO, SAC) and two robust (RARL, RAP) RL agents. In Figure 1, we introduce three types of disturbances, two non-periodic and a periodic one (see the Supplementary Material for more).

White Noise Disturbances We first look at white noise disturbances (Figure 1a), often used to mimic the natural stochastic noise that an agent encounters in the real world. The noise is applied, from zero, at increasing values of standard deviation. In Figure 2, we see very similar low robustness across all control approaches for disturbances on external dynamics, with, as expected, a linear decrease in performance as the noise increases. However, the robust approaches, RARL and RAP, show no significant difference in performance w.r.t. PPO.

For action disturbances, RAP consistently has the highest average normalised return. For observation disturbances, PPO has the highest average normalised return at high levels of disturbances. Overall, the difference across the four approaches is small and they all demonstrate similarly good robustness when white noise is applied to observations or actions.

Step Disturbances Step disturbances (Figure 1b) allow us to see the system's response to a sudden and sustained change. As in all experiments, the disturbance is applied at varying levels, here representing the magnitude of the step. The step occurs two steps into the episode for all runs.

As expected, compared to white noise disturbances, step disturbances have a much greater effect and even low magnitudes result in a large decrease in performance. There are especially steep decreases in average normalised return when the agent can no longer stabilise the cart-pole, e.g., in the second and third plots of Figure 3.

For the step disturbance on external dynamics, there is no unique better controller, with RARL marginally outperforming the others. For actions and observations, PPO again achieves the best overall performance, yielding almost ideal average normalised return up until the step magnitude reaches, for action disturbances, as high as 5 N. SAC's performance is technically higher at low levels of disturbances but fails quickly as the magnitude of the step increases.

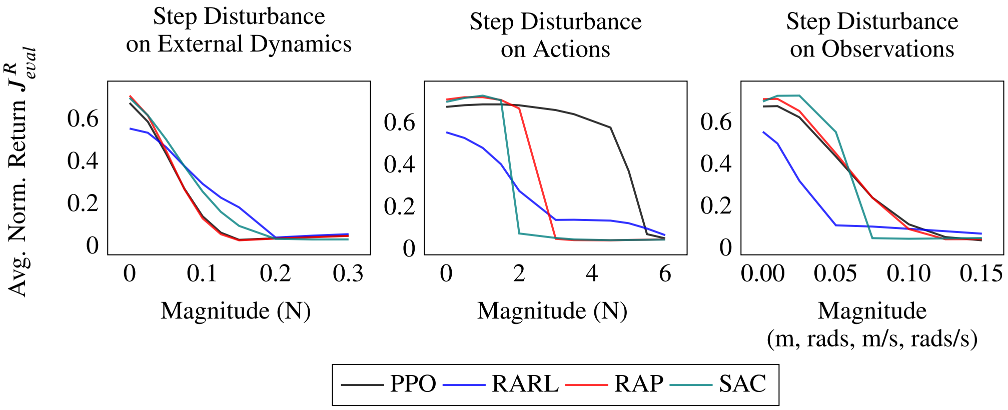

Figure 3: Average normalised return with the injection of step disturbances applied to (left to right) dynamics, actions, and observations on the cart-pole stabilisation task for the four RL agents.

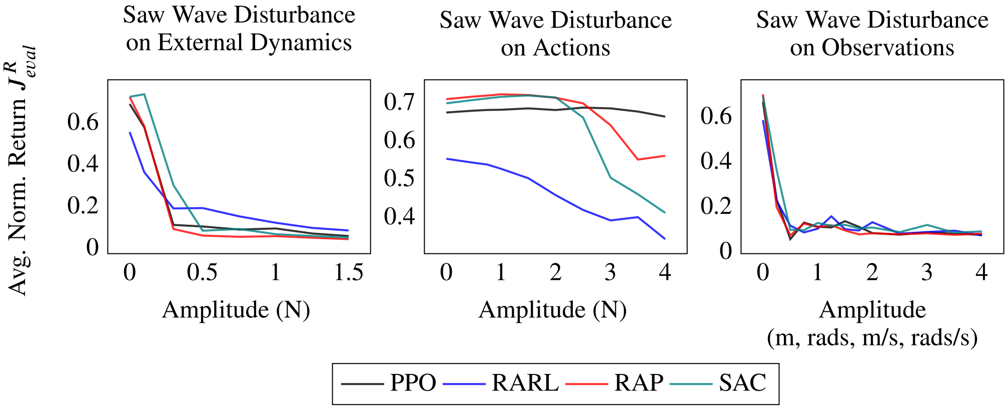

Figure 4: Average normalised return with the injection of sawtooth wave disturbances applied to (left to right) dynamics, actions, and observations on the cart-pole stabilisation task for the four RL agents.

3.2 Periodic Disturbances

Beyond episodic disturbances, we also want to explore the ability of the controllers to deal with periodic disturbances. Such disturbances further challenge the agents as they introduce long-lasting perturbations, that force them to seek robustness in an environment that never behaves as ideal.

Saw Wave Disturbances A saw (or sawtooth) wave (Figure 1c) is a cyclic wave that increases linearly to a set magnitude and instantaneously drops back to a starting point before repeating the cycle. Thus, this disturbance type includes aspects of the step and impulse disturbances, yet it is applied periodically throughout the evaluation episodes.

In Figure 4, the difference in performance between the approaches is less marked (in comparison to the disturbances applied in previous experiments). For disturbances in the dynamics, there is little difference in performance (and low overall robustness) for all control approaches. RARL performs better than the other approaches at low amplitude disturbances. For action disturbances, PPO is the agent that best preserves its average normalised return, while the other approaches, in particular SAC, display lower robustness.

When the policy behaviour of the controllers was re-played, it was evident that RAP and RARL failed more often than PPO and SAC, resulting in a lower average normalised return. When the saw wave disturbance is applied to observations, all approaches have great difficulty stabilising and the average normalised return quickly drops to zero.

4 Conclusions and Outlook

In this article, we presented results that provide insight into the robustness of reinforcement learning, in particular in the context of continuous control. One of our main findings for roboticists is that the RL agents are more susceptible to disturbances injected into dynamics and observations. On the other hand, all agents under test, both vanilla RL agents and robust ones, display some inherent robustness to action disturbances.

As the field of robust reinforcement learning develops, our results indicate that particular care should be dedicated to improving robustness to observation and dynamics disturbances. Nonetheless, building on traditional RL approaches that already demonstrate to generalise well against disturbances may be a promising path for robust robot control using reinforcement learning.

References

[1] Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2017.

[2] Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. Conservative safety critics for exploration. arXiv:2010.14497 [cs.LG], 2021.

[3] Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1), 2022.

[4] Jason J Choi, Donggun Lee, Koushil Sreenath, Claire J Tomlin, and Sylvia L Herbert. Robust control barrier-value functions for safety-critical control. arXiv:2104.02808 [eess.SY], 2021.

[5] J. Collins, S. Chand, A. Vanderkop, and D. Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.

[6] G.E. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*. Texts in Applied Mathematics. Springer New York, 2005.

[7] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax – a differentiable physics engine for large scale rigid body simulation. arXiv:2106.13281 [cs.RO], 2021.

[8] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[9] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[10] C. Karen Liu and Dan Negrut. The role of physics-based simulators in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):35–58, 2021.

[11] Brett T. Lopez, Jean-Jacques E. Slotine, and Jonathan P. How. Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters*, 5(3):1031–1036, 2021.

[12] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU-based physics simulation for robot learning. arXiv:2108.10470 [cs.RO], 2021.

[13] D.Q. Mayne, M.M. Seron, and S.V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[15] Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural Computation*, 17(2):335–359, 2005.

[16] Siddharth Mysore, Bassel Mabsout, Renato Mancuso, and Kate Saenko. Regularizing action policies for smooth control with reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1810–1816, 2021.

[17] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[18] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P. Schoellig. Learning to fly—a Gym environment with PyBullet physics for reinforcement learning of multi-agent quadcopter control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7512–7519, 2021.

[19] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2817–2826. 2017.

[20] Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):253–279, 2019.

[21] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. arXiv:2205.06175 [cs.AI], 2022.

[22] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897. 2015.

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[24] Yunlong Song, Mats Steinweg, Elia Kaufmann, and Davide Scaramuzza. Autonomous drone racing with deep reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1205–1212, 2021.

[25] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. Learning to be safe: Deep rl with a safety critic. arXiv:2010.14603 [cs.LG], 2020.

[26] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.

[27] B. Thananjeyan, A. Balakrishna, U. Rosolia, F. Li, R. McAllister, J. E. Gonzalez, S. Levine, F. Borrelli, and K. Goldberg. Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks. *IEEE Robotics and Automation Letters*, 5(2):3612–3619, 2020.

[28] Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. Robust reinforcement learning using adversarial populations. arXiv:2008.01825 [cs.LG], 2020.

[29] Zhaocong Yuan, Adam W. Hall, Siqi Zhou, Lukas Brunke, Melissa Greeff, Jacopo Panerati, and Angela P. Schoellig. Safe-control-gym: A unified benchmark suite for safe learning-based control and reinforcement learning in robotics. *IEEE Robotics and Automation Letters*, 7(4):11142–11149, 2022.

[30] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

Supplementary Material

In RL, an agent, in our case, a robot, performs an action and receives feedback (reward) from the environment on how well it is doing at the environment's task, perceives the updated state of the environment resulting from the action taken and repeats the process, learning over time to improve the actions it takes to maximise reward collection (and this to correctly perform the task). The resulting behaviour is called the agent's policy and maps the environment's state to actions [26]. While early RL research was demonstrated in the context of grid worlds and games, in recent years, we have seen a growing interest in physics-based simulation for robot learning [7, 10, 18, 5]. For simplicity and reproducibility reasons, however, many of these simulators are still fully deterministic (and prone to be exploited by the agents). In this study, we deliberately inject disturbances at different points of the RL learning and control interaction loop to emulate the conditions an agent might encounter in the real world.

Injecting Disturbances in Robotic Environments

We systematically inject each of the disturbances in Figures 1 and 6 in one of three possible sites: observations, actions, and dynamics of the environment that the RL agent interacts with.

Observation/state Disturbances Observation/state disturbances occur when the robot's sensors cannot perceive the exact state of the robot. This is a very common problem in robotics and is tackled with state estimation methods [1]. In the case of the cart-pole, this disturbance is four-dimensional—as is the state—and is measured in metres in the first dimension, radians in the second, metres per second in the third, and radians per second in the fourth. This disturbance is implemented by directly modifying the state observed by the system.

Action Disturbances Action disturbances occur when the actuation of the robot's motors is not exactly as the control output specifies, resulting in a difference between the actual and expected action. For example, action delays are often neglected or coarsely modeled in simple simulations. In the case of the cart-pole, this disturbance is a one-dimensional force (in Newtons) in the x-direction directly applied to the slider-to-cart joint.

External Dynamics Disturbances External dynamics disturbances are disturbances directly applied to the robot that can be thought of as environmental factors such as wind or other external forces. In the case of the cart-pole, this disturbance is two-dimensional and implemented as a tapping force (in Newtons) applied to the top of the pole.

Reinforcement Learning Agents for Continuous and Robust Control

While some of the most notable results of deep RL control [14] were achieved in the context of discrete action spaces, we focus on actor-critic agents capable of dealing with the continuous actions spaces needed for embodied AI and robotics [20, 16]. Here, we summarise the agent whose results we reported in Section 3.

Proximal Policy Optimisation (PPO) PPO [23] is a state-of-the-art policy gradient method proposed for the tasks of robot locomotion and Atari game playing. It improves upon previous policy optimisation methods such as ACER (Actor-Critic with Experience Replay) and TRPO (Trust Region Policy Optimisation) [22]. PPO reduces the complexity of implementation, sampling, and parameter tuning using a novel objective function that performs a trust-region update that is compatible with stochastic gradient descent.

Soft Actor-Critic (SAC) SAC [8] is an off-policy actor-critic deep RL algorithm proposed for continuous control tasks. The algorithm merges stochastic policy optimisation and off-policy methods like DDPG (Deep Deterministic Policy Gradient). This allows it to better tackle the exploration-exploitation trade-off pervasive in all reinforcement learning problems by having the actor maximise both the reward and the entropy of the policy. This helps to increase exploration and prevent the policy from getting stuck in local optima.

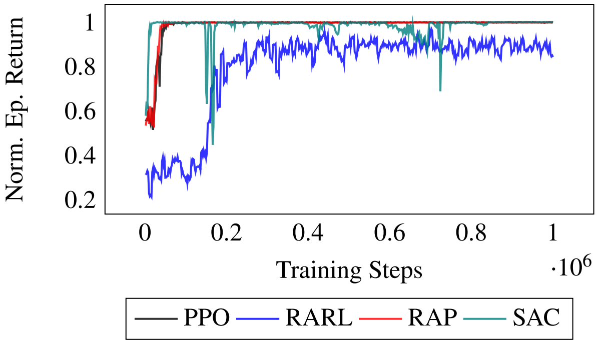

Figure 5: Training curves (# of training steps vs. returns) normalized and averaged over 10 runs in environments without disturbances for all agents on the cart-pole stabilisation task.

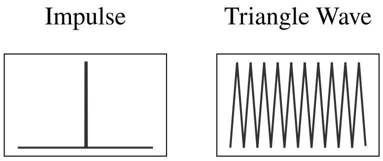

Figure 6: Disturbances injected in the Supplementary Material: impulse and triangle wave.

Robust Adversarial Reinforcement Learning (RARL) Unlike the previous two approaches, RARL [19], as well as the following approach, RAP, are designed to be robust and bridge the gap between simulated results for control and performance in the real world. To achieve this, an adversary is introduced that learns an optimal destabilisation policy and applies these destabilising forces to the agent, increasing its robustness to real disturbances.

Robust Adversarial Reinforcement Learning with Adversarial Populations (RAP) RAP [28] extends RARL by introducing a population of adversaries that are sampled from and trained against. This algorithm hopes to reduce the vulnerability that previous adversarial formulations had to new adversaries by increasing the kinds of adversaries and therefore adversarial behaviours seen in training. Similar to RARL, RAP was originally evaluated on continuous control problems.

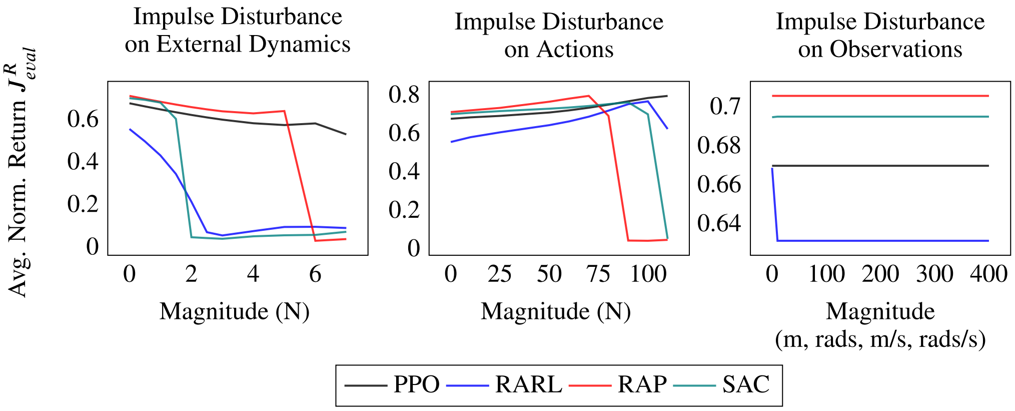

Figure 7: Average normalised return with the injection of impulse disturbances applied to (left to right) dynamics, actions, and observations on the cart-pole stabilisation task for the four RL agents.

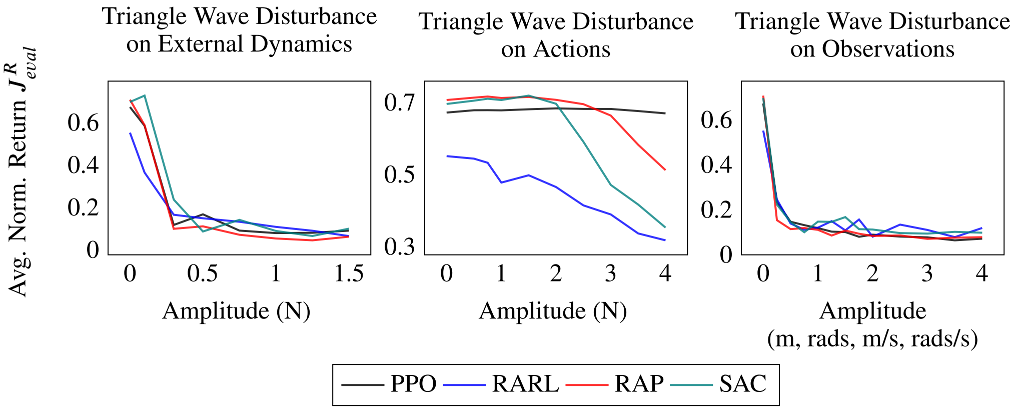

Figure 8: Average normalised return with the injection of triangle wave disturbances applied to (left to right) dynamics, actions, and observations on the cart-pole stabilisation task for the four RL agents.

Impulse Disturbances

Impulse disturbances (Figure 6a) allow us to see the system's response to a sudden, but temporary change. Again, we look at varying levels of the impulse's magnitude to test the controllers' robustness. The width of the impulse is two steps and it is applied two steps into the run for all runs.

In the case of dynamic and action impulse disturbances, the dramatic decrease in performance seen in the previous experiment (with the step disturbances) is just as pronounced. We expected impulse disturbances to be more easily handled than step disturbances, as step responses may require the system to adapt to a new baseline whereas the impulse disturbances' change is only temporary. However, the first two plots in Figure 7 show a dramatic change in average normalised return as the sharp disturbance causes the agent to fail to stabilise. PPO is the most robust to impulse disturbances on external dynamics while RARL displays more robust performance than it did with step disturbances.

For disturbances applied to actions, SAC, PPO, and RARL are able to handle higher magnitudes of impulse disturbance than RAP. On the other hand, the short-lived impulse disturbance on observations does not significantly affect any of the control approaches, even at very high values.

Triangle Wave Disturbance

A triangle wave (Figure 6b) is a cyclic wave that increases linearly to a set magnitude and decreases at the same rate to a starting point before repeating. This disturbance type is very similar to the saw wave disturbance but also acts more similarly to a sinusoidal wave.

Not surprisingly, the results for triangle wave disturbances (Figure 8) are similar to those of the sawtooth wave disturbances. The triangle wave disturbance results in a slightly lower average normalised return than the sawtooth wave disturbances but the relative performance of the control approaches remains the same. SAC performs slightly worse in the case of disturbances applied to dynamics. For disturbances applied to observations, the drop in performance occurs even earlier for all controllers, showing the increased sensitivity to the triangle wave disturbance compared to the sawtooth wave.

Training with Disturbances

In the results presented so far, no additional disturbances were introduced during training. It is natural to wonder whether including disturbances during training (and reducing the distributional shift between train and test scenario) can improve the evaluation performance of the controllers. Disturbances during training—akin to how the RARL and RAP use adversaries to increase their robustness—can potentially lead to the learning of more generalisable policies. In Figure 9, we look at two of the control approaches, PPO, the best performing vanilla RL approach, and RAP, the best performing robust approach, trained with varying levels of white noise (for 1,000,000 steps).

The evaluation/test performance with higher levels of noise is almost always still better when training with low levels of noise, and achieving the best performance when trained with no disturbances. For

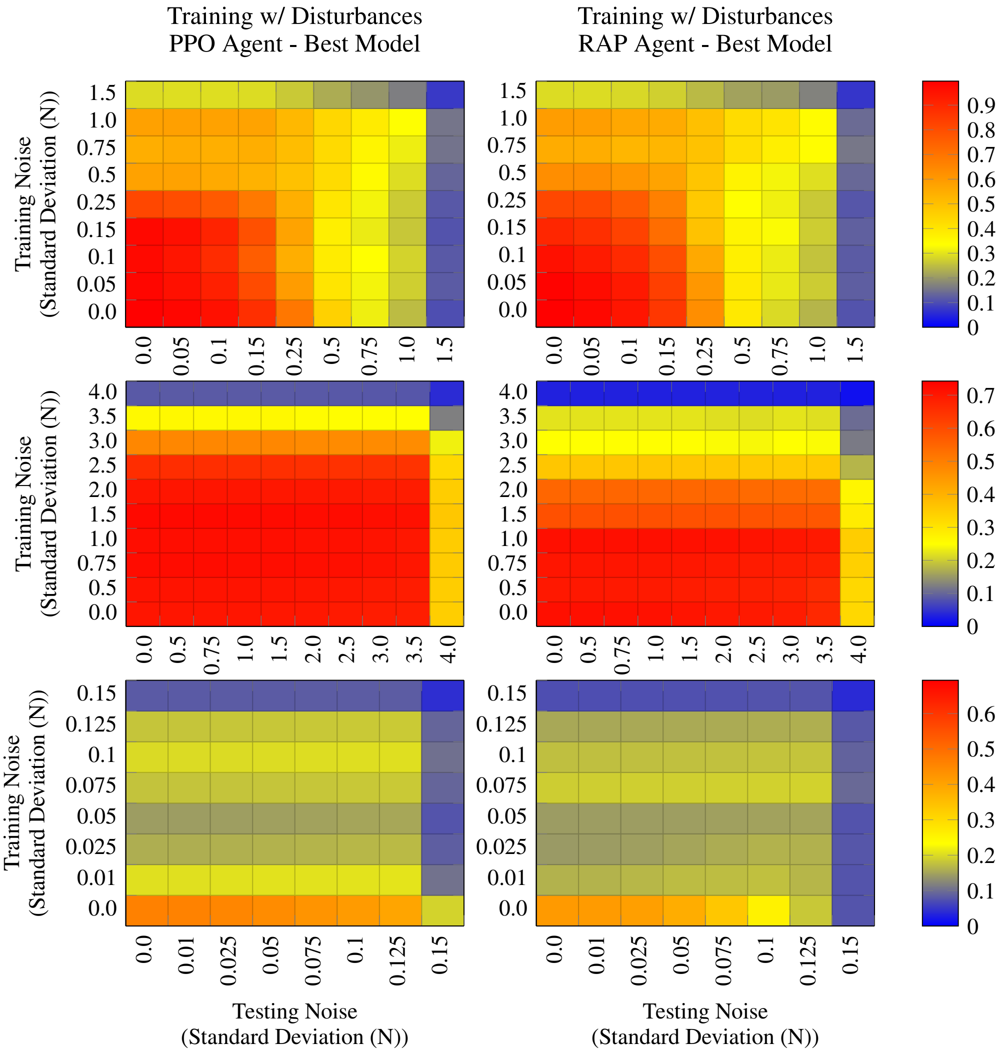

Figure 9: Heat maps of the average normalised return for PPO (left) and RAP (right) trained on (y-axes) and tested (x-axes) with varying levels of white noise on the cart-pole stabilisation task. Rows, from top to bottom, report on disturbances in dynamics, actions, and observations.

external dynamics disturbances, the average normalised return gradually decreases as the training noise is increased. At higher values of training noise, the performance when the levels of testing noise are also higher improves slightly, suggesting there are small improvements. This phenomenon, however, is only visible for disturbances in the dynamics (first row of Figure 9).

For action disturbances, the average normalised return is not affected by increased training noise or testing noise, except at specific, high values where the average normalised return decreases dramatically, showing no obvious performance improvement. For noise added to observations, there is a sudden decrease to nearly zero average normalised return when noise is introduced during training.