Testing Generalizability of Reinforcement Learning with Environmental Stress Tests

Sebastian Sabry
McGill University
sebastien.sabry@mail.mcgill.ca

Basil Jancso-Szabo
McGill University
basil.jancso-szabo@mail.mcgill.ca

Julien Claveau
McGill University
julien.claveau@mail.mcgill.ca

Abstract

This work investigates the generalization capabilities of reinforcement learning (RL) algorithms through stress testing environments during training, extending the empirical framework proposed by Packer u.a. (2019). Rather than evaluating agent performance across fixed train/test environment sets, we introduce dynamic perturbations within the training phase to simulate continuous environmental variability. We implement stress tests by mutating environment parameters in two Arcade Learning Environments: CartPole-v1 and Pacman-ram-v5. Five algorithms are evaluated: Deep Q-Learning (DQN), Actor-Critic (AC), Option-Critic (OC), and the Minorize-Maximization algorithms TRPO and PPO. For each, we conduct a systematic hyperparameter search under normal and stress-tested conditions, introducing reproducible environmental variations such as altered gravity or operational modes. Our methodology is designed to probe algorithmic robustness and encourage a deeper understanding of generalization in RL beyond traditional train/test partitions. Results show that models do tend to perform significantly worse after the environment is perturbed. This is seen most significantly with PPO, which performs best at the baseline without perturbations but is outperformed by simpler methods for a sufficiently large perturbation.

1 Introduction

We extend the work provided in Packer u.a. (2019) in which we define empirical processes to test generalizability for RL algorithms. Previous work created very similar perturbations in environments defining each as a set, in which they can be used to train or validate the algorithm's performance. We hope to extend this by almost creating continuous generalization tests, instead of separating environmental test beds into sets, we introduce perturbations within the environment during training. Appropriately calling such stress tests. We introduce 5 algorithms: Deep-Q-Learning (DQN) Mnih u.a. (2013), policy-gradient method Actor-Critic (AC), an Option-Critic (OC) based architecture Bacon u.a. (2016) and Minorize-Maximization (MM) algorithms TRPO and PPO Schulman u.a. (2017,?). We deploy all algorithms on two separate Arcade Learning Environments (ALE): CartPole-v1 and PacMan. Introducing stress-test functionality in the form of mutations of properties defining each respective environment. This work hopes to investigate the generalization of algorithms within training by introducing perturbations of environments in the form of these stress tests. Our results contribute to the evidence that RL agents can readily overfit to details of the environment, especially with models that have mechanisms for maintaining training stability such as PPO.

41st Conference on Neural Information Processing Systems (NeurIPS 2027).

2 Related Work

The generalizability of reinforcement learning has been thoroughly investigated in previous work and remains a significant challenge, particularly when deploying agents in real-world environments that differ from training settings. Korkmaz (2024) The overfitting of RL policies, in particular deep RL (DRL), is a primary concern, in which the use of deep neural networks as function approximations leads to issues such as estimation biases in state-action value functions. Korkmaz (2021) Some work exists in investigating the role of exploration in overfitting, claiming that simple methods of exploration may be insufficient for complex environments, whereby advanced techniques have been proposed to combat these issues. Weltevrede u.a. (2024) Similarly, in the spirit of combating overfitting of DRL several strategies have been explored, such as data augmentation Laskin u.a. (2020); Kostrikov u.a. (2021), regularization methods Ioffe u.a. (2015); Liu u.a. (2021), and adversarial training Korkmaz (2020); Goodfellow u.a. (2015).

While the issue of generalizability under RL and the potential solutions is heavily studied, there seems to be a lack of work on the empirical processes of defining these generalizability tests. One important work reviews this thoroughly in Packer u.a. (2019). Whereby the authors describe procedures of training on environments under some set, while testing on another set in which the environmental parameters are perturbed by some probability distribution. As such, this work introduces the testbed for an experimental protocol we will investigate in this work.

3 Methodology

To extend the work described in stress-testing in Packer u.a. (2019); Lee u.a. (2020), we must introduce perturbations in the ALE environment as the training is taking place. The Gymnasium API usage for ALE did not allow simple mutations of environments, so we opted to fork the relevant environments and introduce methods to do such.

To evaluate and compare the performance of the chosen algorithms under both standard and perturbed conditions, we conducted a structured hyperparameter search across multiple environments. Hyperparameters were grouped into three categories: algorithm specific, experiment, and stress-test. Algorithm specific hyperparameters were used across all experiments, specific to certain algorithms (e.g. temperature decay for AC, delta for TRPO). These were found from hyperparameter sweeps across the environments with no stress tests. Standard experiment hyperparameters, defined for DQN and OC under default environment settings. For example, learning rate and gamma values. Finally, we defined stress-test hyperparameters used to evaluate the model's robustness under changing conditions. Stress test scenarios were defined with environment-specific modifications:

- **CartPole-v1**: increased gravity
- **Pacman-ram-v5**: altered operational mode, increasing the speed of the agent

These configurations were applied to simulated a changing environment and learning condition to asses the generalizability and robustness of each algorithm.

We used a controlled random seed generator to ensure reproducibility, selecting 10 independent seeds per configuration. For each model:

1. All combinations of hyperparameter values were enumerated.
2. For each combination, the model was trained for 1,000 episodes over 10 independent trials (5 for Pacman-ram-v5 due to constraints in compute).
3. Each model was instantiated with environment-specific configurations and seeded deterministically.
4. After training, episode-wise reward trajectories were saved to disk via JSON format.

Training performance was aggregated across trials and saved per hyperparameter configuration. After training, results were visualized by overlaying multiple models' performance curves for each environment to facilitate direct comparisons.

4 Results

The baseline experiment on CartPole with no stress test is shown Figure 1, showing PPO outperforming all other methods. A sample stress test with gravity increased to 14.8 is shown in Figure 3. Full trials for each variation of gravity can be found in Appendix A. We also include the average performance over the last 100 episodes of each of the methods shown for varying gravity changes in Table 1.

For Pacman-ram-v5, our initial results showed that none of the models were able to learn well in the environment. The results from the stress tests are shown in Appendix A, but there is little analysis of the results since the best-performing models only performed modestly better than the random agent.

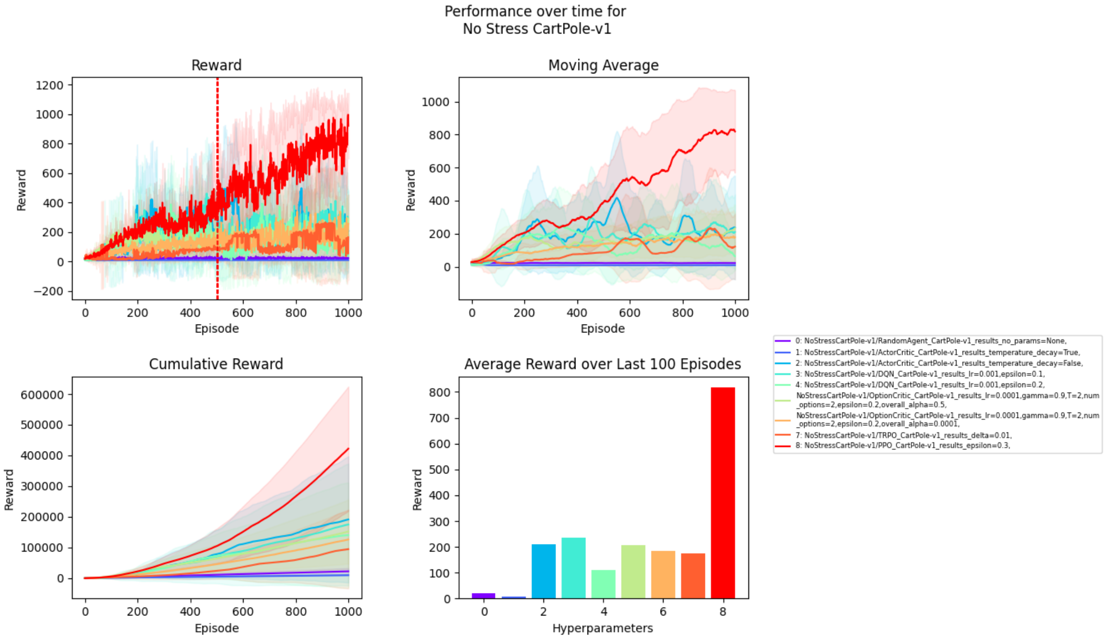

Figure 1: Baseline CartPole trial with no stress test.

Table 1: Average reward per episode for final 100 episodes. We only show the best performing version for models where multiple hyperparameters were tested for the stress test (i.e. Actor Critic with no temperature decay, DQN with $\epsilon = 0.1$, and Option Crtitic with $\alpha = 0.5$).

Algorithms	None	1.5x gravity	2x gravity	2.5x gravity
Random	22.27 ± 3.39	20.76 ± 2.92	19.86 ± 3.05	19.49 ± 2.77
ActorCritic	211.53 ± 55.60	276.81 ± 87.37	153.98 ± 58.43	173.17 ± 49.48
DQN	237.47 ± 67.69	169.82 ± 54.72	154.19 ± 83.77	$\mathbf{179.00 \pm 56.85}$
Option Critic	206.52 ± 32.67	188.04 ± 29.11	171.30 ± 33.44	149.07 ± 26.32
TRPO	174.33 ± 64.18	134.20 ± 30.43	52.49 ± 5.04	44.04 ± 6.28
PPO	$\mathbf{819.05 \pm 83.04}$	$\mathbf{539.34 \pm 122.55}$	$\mathbf{226.42 \pm 42.61}$	128.51 ± 21.37

5 Discussions

The CartPole environment is an easy environment where all models were able to learn and reach a score of around 200 points without any stress test, with PPO obtaining almost 4 times as much as the second best. This can be explained in two parts. First, PPO clips its gradient updates by removing any reward for actions that have already been reinforced too much, stopping the policy from taking over-sized policy updates due to sudden large rewards. Second, PPO gradually updates its policy by stochastic gradient descent between each episode, similar to DQN.

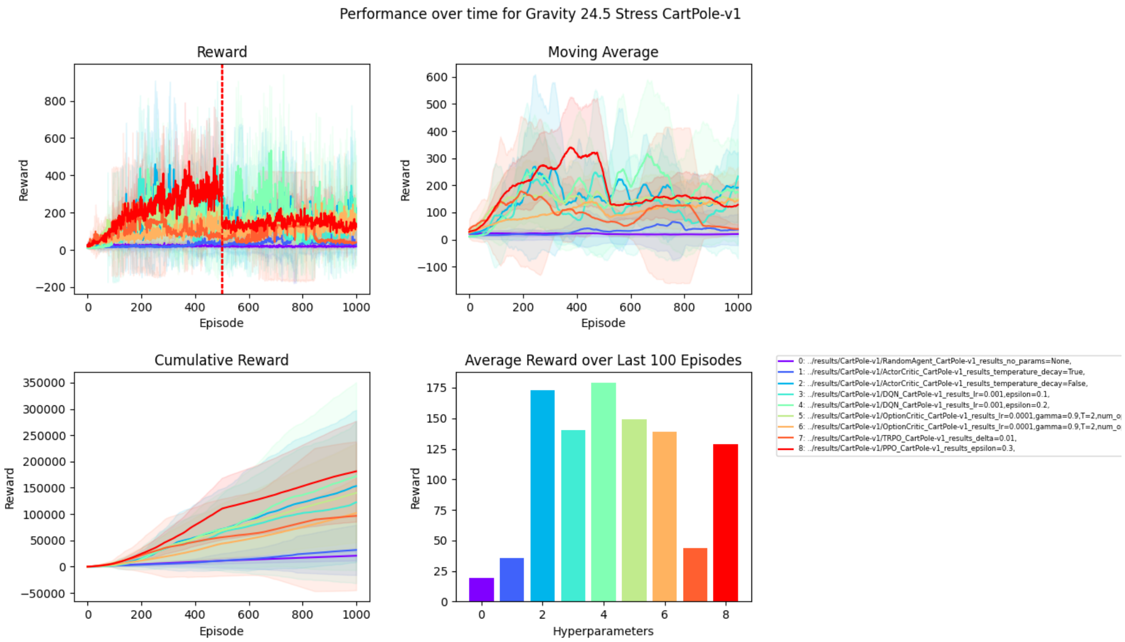

Figure 2: CartPole trial with a stress test of gravity increasing to 24.5 at episode 500.

This seems to increase the precision of the policy updates. These two factors combined allow it to quickly learn its environment without taking too large steps which could lead to catastrophic failure.

However, PPO was also the most affected by the sudden change in the environment caused by our stress tests. It saw a reduction of about 85% of its score when gravity was increased by 2.5x, whereas most other algorithms lost only about 10%-15% of their score. It also scored lower than almost all other algorithms in this stress test. One hypothesis is that by learning a good policy early, it is overfitting to the specific environment, and thus cannot adapt correctly to the new environment. In addition, gradient clipping might prevent it from quickly adapting to the change. DQN, which ends up having the highest score in the hardest stress test, does not overfit before the 500th episode and also does not have the same gradient clipping problem.

6 Conclusion

This project shows that, even under a simple change such as increasing the force of gravity in the Cartpole-v1 environment, RL models struggle to adapt. This change only minorly altered the problem, but we still found that it could have significant impacts on the performance of the models tested, especially for PPO, the top-performing model at baseline.

Future work could explore using different policies for PPO, TRPO, and Actor-Critic to see if better performance could be achieved, especially in the Pacman-ram-v5 environment. Similarly, other methods should be explored for making Option-Critic more stable, as the extremely low learning rate seemed to restrict the performance of the model. It would also be interesting to explore other ALE environments, to see if these models are better suited for different types of problems.

References

Mnih, Volodymyr / Kavukcuoglu, Koray / Silver, David / Graves, Alex / Antonoglou, Ioannis / Wierstra, Daan / Riedmiller, Martin(2013): *Playing Atari with Deep Reinforcement Learning*.

Ioffe, Sergey / Szegedy, Christian(2015): *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*In: Proceedings of the 32nd International Conference on Machine Learning448–456.

Goodfellow, Ian J. / Shlens, Jonathon / Szegedy, Christian(2015): *Explaining and Harnessing Adversarial Examples*.

Bacon, Pierre Luc / Harb, Jean / Precup, Doina(2016): *The Option-Critic Architecture*.

Schulman, John / Wolski, Filip / Dhariwal, Prafulla / Radford, Alec / Klimov, Oleg(2017): *Proximal Policy Optimization Algorithms*.

Schulman, John / Levine, Sergey / Moritz, Philipp / Jordan, Michael I. / Abbeel, Pieter(2017): *Trust Region Policy Optimization*.

Packer, Charles / Gao, Katelyn / Kos, Jernej / Krähenbühl, Philipp / Koltun, Vladlen / Song, Dawn(2019): *Assessing Generalization in Deep Reinforcement Learning*.

Lee, Ritchie / Mengshoel, Ole J. / Saksena, Anshu / Gardner, Ryan / Genin, Daniel / Silbermann, Joshua / Owen, Michael / Kochenderfer, Mykel J.(2020): *Adaptive Stress Testing: Finding Likely Failure Events with Reinforcement Learning*.

Korkmaz, Ezgi(2020): *Nesterov Momentum Adversarial Perturbations in the Deep Reinforcement Learning Domain*.

Laskin, Michael / Lee, Kimin / Stooke, Adam / Pinto, Lerrel / Abbeel, Pieter / Srinivas, Aravind(2020): *Reinforcement Learning with Augmented Data*.

Kostrikov, Ilya / Yarats, Denis / Fergus, Rob(2021): *Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels*.

Korkmaz, Ezgi(2021): *Investigating Vulnerabilities of Deep Neural Policies*.

Liu, Zhuang / Li, Xuanlin / Kang, Bingyi / Darrell, Trevor(2021): *Regularization Matters in Policy Optimization*.

Weltevrede, Max / Kaubek, Felix / Spaan, Matthijs T. J. / Böhmer, Wendelin(2024): *Explore-Go: Leveraging Exploration for Generalisation in Deep Reinforcement Learning*.

Korkmaz, Ezgi(2024): *A Survey Analyzing Generalization in Deep Reinforcement Learning*.

A Full Results

We show the results from trials where gravity is increased to 14.8 in Figure 3 and 19.6 in Figure 4 after episode 500. Figure 5 shows the results from our single trial with Pacman-ram-v5, showing that no model was able to significantly outperform the Random Agent.

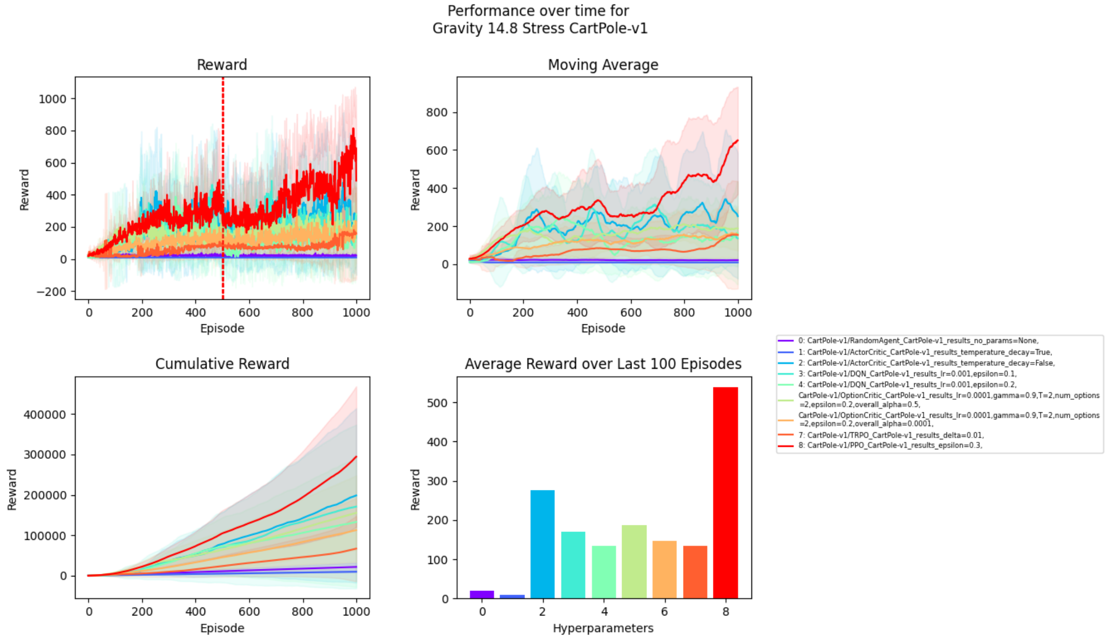

Figure 3: Trial with a stress test of gravity increasing to 14.8 at episode 500.

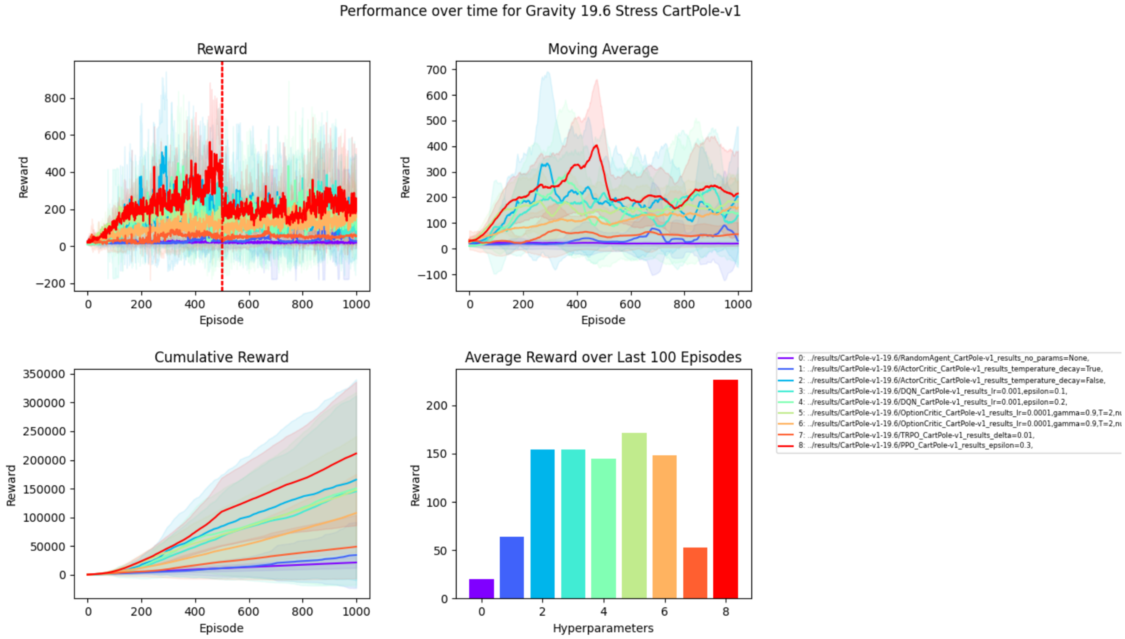

Figure 4: Trial with a stress test of gravity increasing to 19.6 at episode 500.

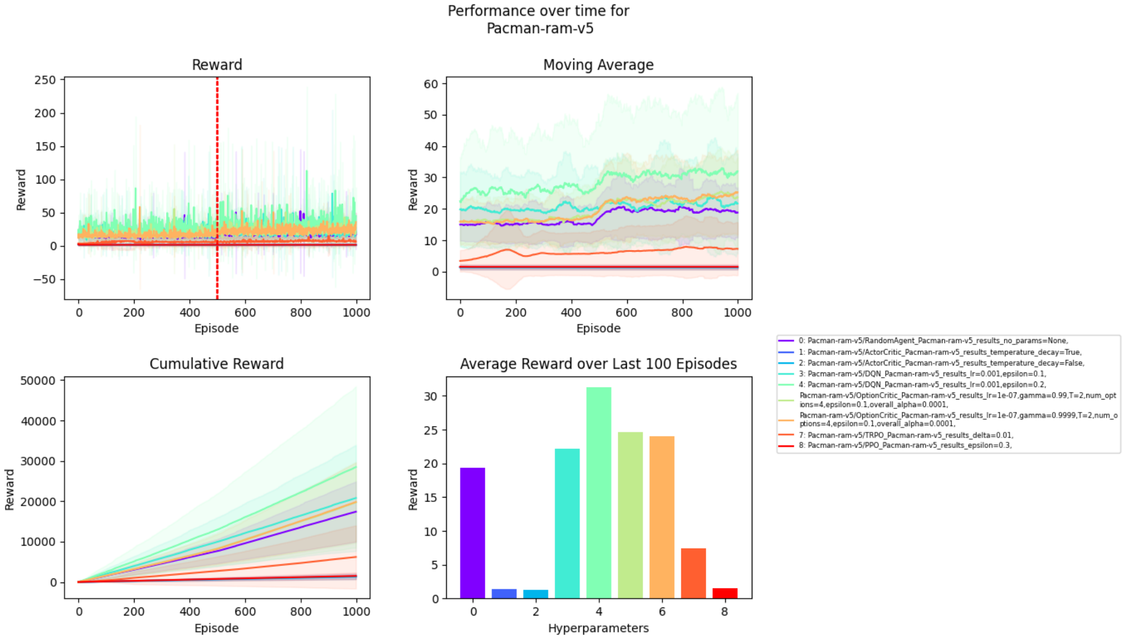

Figure 5: Pacman trial with a stress test of switching to game mode 4 at episode 500.