Analysis of Safety Guarantees over Evolving Controllers

Tanmay Bhaskar Khandait, [1] **Kishore Ramesh,** [2] **Vennela Kudala** [3]

[1] 1219385830, [2] 1229558761, [3] 1232668179
School of Computing and Augmented Intelligence, Arizona State University
[1] tkhandai@asu.edu, [2] krames19@asu.edu, [3] vkudala@asu.edu

Introduction

Self-driving cars, robots and drones are becoming more and more common in various fields, and these systems often use complex controllers that are trained using reinforcement learning algorithms. However, ensuring the safety and reliability of these controllers is a major challenge, especially in dynamic and uncertain environments.

One way to tackle this challenge is to analyze the safety guarantees of the controllers by verifying their behavior against formal specifications. Signal Temporal Logic (STL) is a commonly used formalism for specifying desired properties of systems, such as safety, stability, and robustness. To estimate the likelihood of violating an STL specification by exploring the input space of the controller, falsification techniques like the PART-X algorithm are employed.

This study investigates how an evolving controller behaves against a specification and provide probabilistic guarantees on it. Evolving controllers with respect to this study refers to the intermediary controllers that can be generated while training an Reinforcement Learning (RL) agent to learn a certain policy. A simple example of evolving controllers could be controllers saved at every few iterations of the training algorithm until it terminates. On the other hand, guarantees on the violation (or satisfaction) of a certain behavior is generated using PART-X (Pedrielli et al. 2023) algorithm. By capturing the safety guarantees at different stages of the training process, this study aims to provide insights into the trade-offs between performance and safety and guide the training process to improve the safety of the final controller. This study is performed on the Cart-Pole environment (Barto, Sutton, and Anderson 1983) where the controller is generated using the PPO algorithm and the guarantees are generated under two different settings. This work also proposes a general framework for analyzing the guarantees of evolving controllers using black-box falsification techniques.

Related Work

RL is a popular approach for training controllers in various domains, including robotics and control systems. Algorithms such as DQN (Mnih et al. 2015), TRPO (Schulman et al. 2015), and PPO (Schulman et al. 2017) have been successful in training controllers for complex tasks. Formal verification techniques, such as model checking and theorem proving, are often used to verify the correctness of systems against specifications. Various tools like Breach (Donzé 2010) and S-TaLiRo (Annpureddy et al. 2011) address the issue of finding counterexamples against a specification. While most of the methods try to use ideas of falsification (Wang, Nair, and Althoff 2020) or robustness as a reward function (Kapoor, Balakrishnan, and Deshmukh 2020) to train the agent and generate policies, to the best of our knowledge, none of the work evaluate how the falsifying sets look like when a controller evolves.

The PART-X algorithm is a partitioning-based falsification technique that estimates the volume of the input space that violates an STL specification. By providing probabilistic guarantees on the likelihood of encountering falsifying behaviors, PART-X can be used to analyze the safety guarantees of evolving controllers during the training process.

Background

This section provides an overview of the background needed to understand our approach, explaining the PART-X algorithm.

PART-X

Given an STL specification φ, identifying the set of counterexamples can be formulated to estimating the zero-level set of the robustness function associated with the STL. The PART-X algorithm (Pedrielli et al. 2023), approximates the zero-level set with probabilistic guarantees to characterize its volume. The approach is a partitioning-based algorithm that dynamically partitions the input space using different Gaussian processes for each sub-region of the partition as a surrogate for the robustness function. By leveraging these local surrogate models (Gaussian processes), the PART-X algorithm constructs a predictor for the robustness of non-evaluated inputs. This prediction is used to classify a region as positive (satisfying the requirement), negative (violating), or remaining. If a region is classified as remaining or it is reclassified, it is partitioned in the subsequent iteration, provided that the volume after partitioning is greater than a user-specified minimum. Otherwise, the sub-region is sampled but not branched. The algorithm concludes when the budget is exhausted or when all sub-regions reach the user-defined minimum volume. At this point, PART-X returns estimates of the likelihood of encountering a falsifying

input when none are found, and it determines the normalized volume of falsifying sets (0-level sets).

Technical Details

Approach

Our approach to verifying evolving controllers consists of two phases. The first phase involves the generation of the controllers. The second phase involves generating guarantees on the controller using PART-X.

Generating the Controllers This stage involves the generation of controllers. Since we are dealing with evolving controllers, we assume that there exists a functionality to save the controller at various intermediary stages in addition to the final controller generated at the termination of the algorithm. It is important to note that the algorithm that used for generation of the controller could be anything. Once these controllers are generated, they are converted to a black box system such that it takes in a certain input and produces a trajectory of states that the system was in.

This study is performed on the Cart-Pole (Barto, Sutton, and Anderson 1983) from the OpenAI Gym, where a pole is attached to a cart with an un-actuated joint. The cart moves over a frictionless surface. The goal of a controller is to balance the pole by applying forces in the left or right direction on the cart. There are two sets of inputs called the environment variables and state variables. The environment variables include the mass of the pole, the length of the pole, and the force magnitude that is applied to the cart, which can also be understood as variables that affect the underlying dynamics of the Cart-Pole environment. The state variables include the position of the cart, velocity of the cart, angle of the pole from the vertical and the angular velocity of the pole. These variables could be understood as the observation of the pole and cart at different time steps.

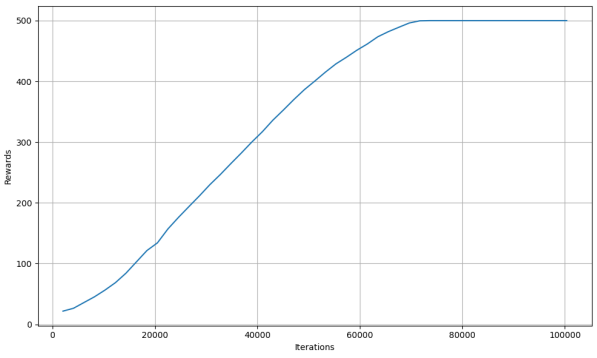


Figure 1: Rewards while training the policy using PPO.

The policy for the agent is learned using the PPO algorithm. Specifically, the environment variables are fixed and the PPO algorithm is allowed to learn a policy such that the pole is eventually balanced on top of the cart. The PPO algorithm is allowed to run for 100000 iterations and an intermediate policy is stored every 10000 iterations. The training curve of the graph is shown in Fig. 1.

Generating Guarantees using PART-X Once the black box system is generated, a specification φ is identified that the system is expected to follow. These specifications could address various properties including safety, invariance, and stability. PART-X is then applied to generate the guarantees in terms of falsification volume (the ratio of the volume of the input space that violates the specification to the total volume of the input space).

For the first experiment, we run PART-X on a 3-dimensional case. The initial position of the cart x, the initial velocity of the cart v, the initial angle of the pole form the vertical θ, and the angular velocity $\dot{\theta}$ are all fixed to 0.001. The environment variables over which the guarantees are generated are sampled from $[0.05, 0.15] \times [0.4, 0.6] \times [0, 10]$. The first one refers to the mass of pole M, the second refers to the length of the pole l, and the third one refers to the initial force magnitude F.

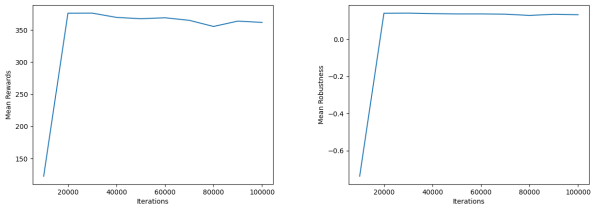


Figure 2: Mean Rewards and Robustness of intermediate policies using 10000 different observations for the 3D case.

Specification: The specification for the controller is a stability requirement shown below:

$$\varphi \equiv \Diamond\Box(-1 \leq x \leq 1) \wedge \Diamond\Box(-1 \leq \theta \leq 1) \wedge \Diamond\Box(-1 \leq p \leq 1)$$

This specification says that eventually always the position of the cart should be between -1 and 1 and eventually always the angle of the pole from the vertical should be between -9° and 9° and eventually always the momentum of the cart should be between -1 and 1.

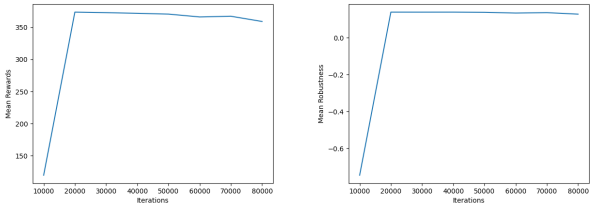


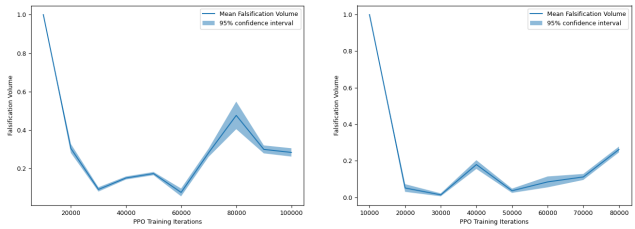
Figure 3: Mean Rewards and Robustness of intermediate policies using 10000 different observations for the 7D case.

For the second experiment, the PART-X algorithm is run on varying initial observations and environment variables. The search space is a 7-dimensional search space defined as $[-2, 2] \times [-0.05, 0.05] \times [-0.2, 0.2] \times [-0.05, 0.05] \times$

Name	Phase 1	Phase 2
Tanmay Khandait	Looked at Non-Linear Gaussian Process Regression (GPR) and Deep GPR Theory.	Extending Deterministic Part-X to Stochastic Part-X - Work in Progress. Integrating Psy-TaLiRo (our Tool) with the gym environment. Analysing results and generating plots.
Kishore Ramesh	Running and setting up the implementation of reference papers.	Setting up Gym Black-Box Function similar to the reference paper - but ensuring reproducibility of Falsifying points.
Vennela Kudala	Looked at implementation of Multi Fidelity BO from BOTorch examples. Main focus was on using the GPU functionality to run our experiments.	Set up Experiments on the Cluster while looking for key results. Running the 7D experiments to ensure we have fair and reproducible results.
Overall Achievements	We found a bug with the paper which could probably be due to oversight on how gymnasium and Stable Baselines environments work.	Ran a total of 198 experiments on deterministic version of the Problem in a 3D and 7D space.

Table 1: Summary of individual contributions and key achievements from both the phases of the project.

$[0.05, 0.15] \times [0.4, 0.6] \times [0, 10]$. The search space corresponds to the initial position of the cart x, the initial velocity of the cart v, the initial angle of the pole from the vertical θ, the initial angular velocity $\dot{\theta}$, the mass of pole M, the length of the pole l, and the initial force magnitude F respectively. The specification is the same as that for the 3D case.



(a) Falsification Volumes and the lower and upper confidence bounds on the 3D case.

(b) Falsification Volumes and the lower and upper confidence bounds on the 7D case.

Figure 4: Probabilistic estimates of falsification on intermediate policies

Individual Contributions

In the first phase of the project, we were working on developing and integrating multi-fidelity BO with PART-X, and our reference paper was based on (Shahrooei, Kochenderfer, and Baheri 2023). However, we pivoted after the proposal review to the current project. Individual contributions by the individual team members are summarized in Table 1.

Results and Observation

Fig. 2 shows mean rewards and mean robustness against the specification of the intermediate policies generated every 10000 iteration are computed over 10000 samples of the 3-dimensional search space. Similarly, the mean rewards and mean robustness for the 7-dimensional case are plotted in Fig. 3. It is clear that the stability specification we consider and the rewards earned by the agent are correlated to each other.

The PART-X algorithm is run for 10 times using a maximum budget of 2000 samples. The falsification volumes and the associated 95% lower and upper confidences are plotted in Fig. 4a and Fig. 4b for the 3D and the 7D cases respectively. We observe that as the training converges (Fig. 1), the falsification guarantees of the stability specification keeps increasing. The lowest falsification volume is achieved at 60000 and 30000 iterations for the 3-dimensional and the 7-dimensional cases respectively, after which it keeps increasing. This coupled with almost no deviation in the mean rewards and robustness suggest that the agent can balance the pole but the stability specification is being violated. This hints that there might be over-fitting happening while we train the controller on the fixed environment variables.

Safety Dimensions Addressed

With respect to the safety dimensions, this work probably address the "Agent's Behavior Synthesis". This is because our work studies how the agents behavior evolves with respect to certain specifications while it is still in the training phase.

Future Work and Conclusion

In this work, we tried to study the probability of the evolving controllers that were generated during training phase following a user-defined specification. While this work is not complete, some of the key observations are that probabilistic guarantees of following (or falsifying) a specification and the training of the controller might not be dependent on each other. In addition, there might be a way to use these guarantees on the intermediate controllers to focus the training of controllers. Currently, we perform experiments in the RL framework and we chose a scenario where the specification and rewards reflect each others behavior directly. The future work would involve looking at scenarios where the specification robustness and the rewards generated by the environment are orthogonal to each other. There is also a significant benefit in developing a stochastic version of the PART-X algorithm which can be utilized to perform a more robust analysis. Please visit https://github.com/kishore-05/Group_9_AI_Safety_Project for source code.

References

Annpureddy, Y.; Liu, C.; Fainekos, G.; and Sankaranarayanan, S. 2011. S-taliro: A tool for temporal logic falsification for hybrid systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 254–257. Springer.

Barto, A. G.; Sutton, R. S.; and Anderson, C. W. 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5): 834–846.

Donzé, A. 2010. Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *Computer Aided Verification: 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings 22*, 167–170. Springer.

Kapoor, P.; Balakrishnan, A.; and Deshmukh, J. V. 2020. Model-based reinforcement learning from signal temporal logic specifications. *arXiv preprint arXiv:2011.04950*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Pedrielli, G.; Khandait, T.; Cao, Y.; Thibeault, Q.; Huang, H.; Castillo-Effen, M.; and Fainekos, G. 2023. Part-x: A family of stochastic algorithms for search-based test generation with probabilistic guarantees. *IEEE Transactions on Automation Science and Engineering*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shahrooei, Z.; Kochenderfer, M. J.; and Baheri, A. 2023. Falsification of Learning-Based Controllers through Multi-Fidelity Bayesian Optimization. In *2023 European Control Conference (ECC)*, 1–6.

Wang, X.; Nair, S.; and Althoff, M. 2020. Falsification-Based Robust Adversarial Reinforcement Learning. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 205–212.