Frameworks for Safe Reinforcement Learning

Kiran Sthanusubramonian
ksthanus@asu.edu
School of Computing & Augmented Intelligence
Arizona State University

Introduction & Problem Statement

Reinforcement Learning (RL) is a powerful class of Machine Learning (ML) algorithms that has seen application in various fields such as robotics, game-playing (arcade games, Chess, Go, etc.), and autonomous driving.

The criticality of safety in Reinforcement Learning has been an important research area in RL over the last few years. Safety in RL is paramount due to the potential risk of harm that arises from the interaction between RL agents and their environments. Consequently, if an RL agent makes unsafe decisions, it can lead to tangible consequences such as physical harm, damage to property, or financial loss. Moreover, safety considerations in RL extend beyond immediate risks to encompass broader challenges such as generalization issues, adversarial attacks, and ethical concerns. RL algorithms trained in simulated environments may fail to generalize effectively to real-world scenarios, leading to unexpected and unsafe behavior, which would hamper the widespread adoption of RL algorithms.

Hence, it is imperative to create reliable frameworks and infrastructures for the development of Safe Reinforcement Learning algorithms. Such infrastructures will cater to a large audience: newcomers and seasoned researchers in RL alike will have access to a wide array of pre-existing algorithms, environments, and benchmarking capabilities to ensure the transparency, interpretability, verification, and validation of newly designed safe RL algorithms.

This project will implement and test two newly proposed frameworks that can potentially be used for a wide setting of Safe Reinforcement Learning experiments. The primary motivating publication for this project is OmniSafe (Jiaming Ji 2023). OmniSafe provides a comprehensive architecture for Safe Reinforcement Learning, focusing on providing a highly modular interface with high-performance parallel computing capabilities. Furthermore, this infrastructure implements various algorithm types, including On-Policy, Off-Policy, Offline, and Model-Based RL algorithms.

The second framework/infrastructure of focus is implementing benchmarks for Offline Safe Reinforcement Learning (OSRL) proposed in (Liu et al. 2023). This publication covers two significant aspects of safe offline Reinforcement Learning: constructing datasets for offline safe RL (in the same vein as D4RL (Fu et al. 2020)) and tailoring a benchmarking platform for offline RL algorithms on these datasets. Deep-diving into the implementation details of both these publications will provide a clear understanding of the current state of safe Reinforcement Learning infrastructures present for research.

Related Work

Our primary literature survey was extended mainly to four different existing publications/frameworks. Developed by OpenAI, the Safety Starter Agent (Ray, Achiam, and Amodei 2019) is an RL framework designed to enable researchers to explore safety techniques in RL tasks. To the best of our knowledge, this is the first framework that incorporates a collection of safety algorithms, including constraint optimization, reward shaping, and safe exploration techniques. It also provides a set of benchmark environments designed to evaluate the safety and robustness of RL algorithms. However, since 2020, the maintenance of this project has been discontinued, and its backend framework is built using TensorFlow v1. This highlights another significant requirement for safe RL infrastructures: maintainability.

The second publication/framework of note is the Bullet Safety Gym. (Gronauer 2022). This framework primarily implements agents from the MuJoCo domain for four different action types. Although this framework benchmarks many safe RL algorithms, it does not extend to other famous RL simulation domains. Coming from the robotics angle, we also conducted research on the Safety Control Gym (Yuan et al. 2021), a unified suite used for benchmarking control and reinforcement learning algorithms built on top of PyBullet for robot agents.

The most significant contribution to this problem space is Safety Gymnasium (2023((Ji et al. 2023). It provides various tasks, each presenting unique challenges and safety

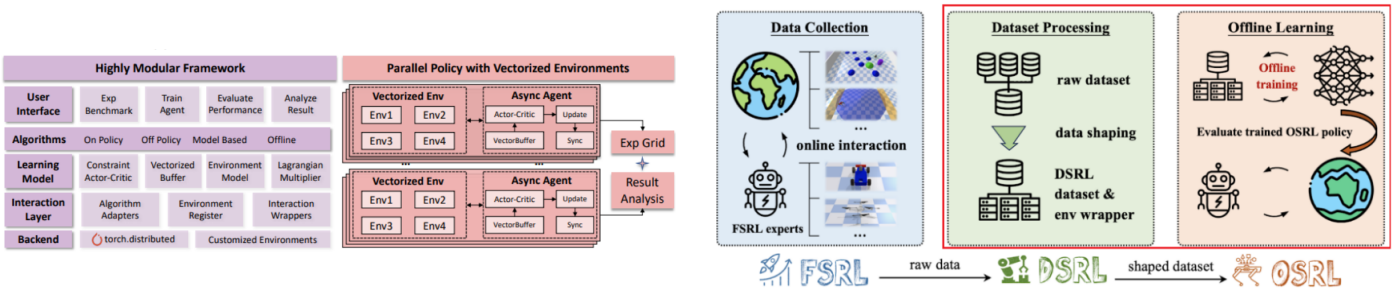


Figure 1: Overview of OmniSafe (left) & OSRL (right) architectures

considerations, ensuring a comprehensive evaluation of RL algorithms. SafetyGymnasium seamlessly integrates with popular RL libraries, including OpenAI Gym and TensorFlow, facilitating experimentation and assessment across different frameworks. This framework parallels OmniSafe, the principal publication leveraged in this project. OmniSafe combines several frameworks, including Safety Gymnasium and Safety Bullet Gym. It is also worth noting that the benchmarking results provided in the Safety Gymnasium GitHub page credit OmniSafe as the benchmarking platform used.

Technical Details

Overall Technical Approach

Stage one of this project involved a thorough background study of classical Reinforcement Learning algorithms implemented in the mentioned publications. The primary focus was Online and Offline Safe RL Algorithms (the algorithms primarily implemented and benchmarked). As part of my work, we implemented the Policy Gradient algorithm to get a complete end-to-end understanding of coding an RL algorithm.

The second stage of this project involved using and benchmarking Online Safe RL algorithms using OmniSafe. OmniSafe provides a highly modular framework for each component required for algorithm implementation and benchmarking, including Safe RL algorithms (using actor, critic, and actor-critic implementations). The actor-critic implementation usually involves deep neural networks that parameterize the actor and critic accurately for complex environments and tasks. For this, a thorough understanding and comparison of several online safe RL algorithms was required. The primary online algorithms benchmarked in this project include Trust Region Policy Optimization (TRPO), Proximal Policy Optimization (PPO), Conservative Policy Optimization (CPO), and Provably Convergent Policy Optimization (PCPO), along with the Policy Gradient implementation method mentioned. We needed to go through how first-order and second-order approximations were used in developing these algorithms, as this would be foundational for a deeper understanding of RL algorithms.

Although OmniSafe extends its infrastructure for Offline RL algorithms as well, we opted to trial OSRL, which provides a complete end-to-end implementation for training safe Offline RL Algorithms, starting from Dataset Generation to Algorithm Benchmarking (Figure 1). This was the final stage of our project. We attempted to implement two Offline RL algorithms based on Deep Q-learning: Batch-Constrained Q-learning (BCQ) and Constrained Penalty Q-learning (CPQ).

The primary advantage of both the above infrastructures mentioned is the large number of in-built environments (based on SafetyGymnasium, MuJoCo, Atari games, etc.) and the capability of customizing relevant environments for different classes of problems built for Python. In our work, we provide benchmarking for three different environments for the Online Algorithms and one for the Offline Algorithms. Safety Point Goal, Safety Ant-Velocity, and Safety Car Circle (used for both Online & Offline) are the three environments.

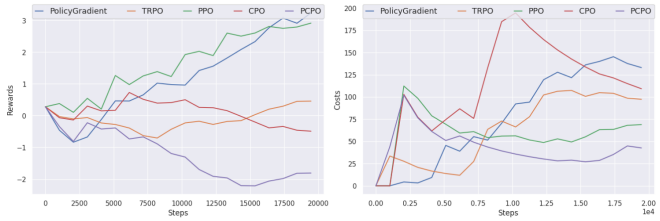
We attempted to provide normalized return and cost metrics for each of the algorithms mentioned; however, due to difficulty in reconciling the metrics provided in OSRL, we currently only provide regular return and cost as evaluation metrics.

Project Contributions

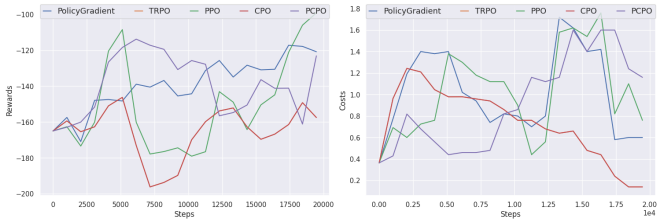
This was a solo project. I intended to use it as a stepping stone to deeper contributions in RL. I found this project to be extremely challenging, and looking back, I feel I should have conducted this work in a 2-member team. However, I am proud of my learning and what I managed to deliver for this project. Learning more about end-to-end RL infrastructures, from Dataset Preparation and Preprocessing to Algorithm Development, has enhanced my repertoire as an AI/ML Engineer - which was my primary goal for this course.

Results

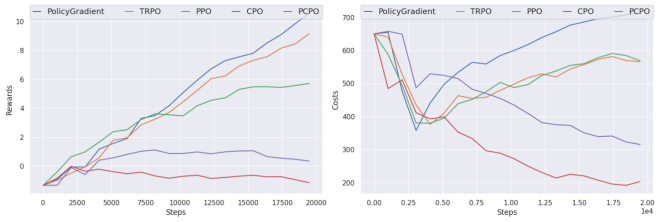
For each of our Online-based RL algorithm benchmarking experiments (based on OmniSafe), we ran each algorithm for three different seeds and provided the average return and cost per time step. Each algorithm was run for 20000 time steps (interactions with the environment). The results are presented in Figure 2. The entire code for benchmarking and experimentation was implemented as a Jupyter Notebook



(a) Environment: Safety Point-Goal



(b) Environment: Safety Ant-Velocity

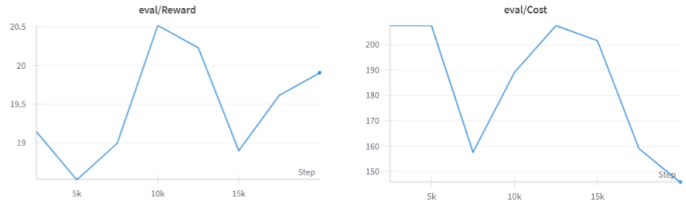


(c) Environment: Safety Car-Circle

Figure 2: Online Safe RL Algorithms Benchmarking

and ran on Google Colaboratory (CoLab). Due to GPU accelerator constraints, we were only able to simulate 20000 time steps for each algorithm. For the Online-RL algorithm benchmarking case for the three environments mentioned, we successfully replicated the original results published in OmniSafe benchmarking tests. Overall, it was noticed that the first-order approximation-based algorithm in Proximal Policy Optimization (PPO) reliably gave the best reward-to-cost ratio for the environments we benchmarked upon.

We ran into several implementation issues for the Offline-based RL algorithm benchmarking experiments (based on OSRL). For understanding the data collection stage, we easily managed to understand the workflow for creating and hosting offline RL datasets (similar to D4RL). However, when it came to the algorithm execution stage, we encountered several minor bug fixes to complement the BCQ and CPQ algorithms. There was no quality documentation for the code repository we were using, and the metrics generation and visualization techniques used were confusing and time-consuming to run for larger experiments. We managed to get the BCQ implementation working. The CPQ algorithm was eventually bug-free and could be executed but gave poor results (overall cost throughout the algorithm was 0). We provide the BCQ Offline Algorithm results for the Safety Car Circle environment in Figure 3(a).



(a) Environment: Safety Car-Circle

Figure 3: Offline Safe RL Algorithms Benchmarking

Comparing the performance of Offline Algorithms (OSRL) and Online Algorithms (OmniSafe), we can clearly see that the offline algorithm's overall reward and cost converge faster and provide better numbers. This is to be expected in Offline Q-learning-based simulations with dedicated datasets, and we intend to further corroborate this statement as an extension to this project in the near future.

Safety Dimensions Addressed

The safety dimensions addressed in this project are as follows:

1. **AI Objective vs. AI Behaviour**: The implementation of safe RL algorithms with well-specified constraints automatically comes under the purview of AI Objective vs. AI Behaviour, as algorithm design is a translation between an objective and the overall behavior of an agent.

From the work that was achieved, there is an argument for the dimension of **Computed Behaviour vs. Real Outcome** also being addressed, but I do not think simulations of computed behavior can constitute under the realm of Real Outcomes.

Conclusions and Future Work

The primary conclusion of this project is that the growing need for reliable, extensible, modular, maintainable, and highly reproducible frameworks/infrastructures for safe Reinforcement Learning is going to be the bedrock for developing new safe RL algorithms and related research. OmniSafe currently fits the bill for each of the adjectives used, and it's very well-documented and suited for beginners and experts alike. In contrast, OSRL provides its own plus points and ideas required for benchmarking Offline Safe RL algorithms with a rich dataset collection process, although the algorithm benchmarking and documentation were vague and tough to implement.

As part of future work, OmniSafe (if it gains sufficient traction) can become the state-of-the-art infrastructure used for benchmarking and implementing safe RL algorithms. An exciting possibility would be to leverage the use of the dataset generation capabilities provided in OSRL and implement the same in OmniSafe's Offline RL Algorithm Suite.

References

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219.

Gronauer, S. 2022. Bullet-Safety-Gym: A Framework for Constrained Reinforcement Learning. Technical report, mediaTUM.

Ji, J.; Zhang, B.; Zhou, J.; Pan, X.; Huang, W.; Sun, R.; Geng, Y.; Zhong, Y.; Dai, J.; and Yang, Y. 2023. Safety Gymnasium: A Unified Safe Reinforcement Learning Benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jiaming Ji, B. Z. J. D. X. P. R. S. W. H. Y. G. M. L. Y. Y., Jiayi Zhou. 2023. OmniSafe: An Infrastructure for Accelerating Safe Reinforcement Learning Research. *arXiv preprint arXiv:2305.09304*.

Liu, Z.; Guo, Z.; Lin, H.; Yao, Y.; Zhu, J.; Cen, Z.; Hu, H.; Yu, W.; Zhang, T.; Tan, J.; and Zhao, D. 2023. Datasets and Benchmarks for Offline Safe Reinforcement Learning. arXiv:2306.09303.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking Safe Exploration in Deep Reinforcement Learning.

Yuan, Z.; Hall, A. W.; Zhou, S.; Brunke, L.; Greeff, M.; Panerati, J.; and Schoellig, A. P. 2021. safe-control-gym: a Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning. arXiv:2109.06325.

Appendix

Code Repository Link: https://github.com/kiran-asu5115/Safe-Reinforcement_Learning