Journal of Machine Learning Research 25 (2024) 1-6 Submitted 5/23; Revised 9/24; Published 9/24

OmniSafe: An Infrastructure for Accelerating Safe Reinforcement Learning Research

Jiaming Ji* JIAMG.JI@GMAIL.COM
Jiayi Zhou* GAIEJJ@OUTLOOK.COM
Borong Zhang* BORONGZH@GMAIL.COM
Juntao Dai JTD.ACAD@GMAIL.COM
Xuehai Pan XUEHAIPAN@PKU.EDU.CN
Ruiyang Sun SUN_RUIYANG@STU.PKU.EDU.CN
Weidong Huang BIGEASTHUANG@GMAIL.COM
Yiran Geng GYR@STU.PKU.EDU.CN
Mickel Liu MICKELLIU7@GMAIL.COM
Yaodong Yang† YAODONG.YANG@PKU.EDU.CN
Institute for Artificial Intelligence, Peking University, China

Editor: Joaquin Vanschoren

Abstract

AI systems empowered by reinforcement learning (RL) algorithms harbor the immense potential to catalyze societal advancement, yet their deployment is often impeded by significant safety concerns. Particularly in safety-critical applications, researchers have raised concerns about unintended harms or unsafe behaviors of unaligned RL agents. The philosophy of safe reinforcement learning (SafeRL) is to align RL agents with harmless intentions and safe behavioral patterns. In SafeRL, agents learn to develop optimal policies by receiving feedback from the environment, while also fulfilling the requirement of minimizing the risk of unintended harm or unsafe behavior. However, due to the intricate nature of SafeRL algorithm implementation, combining methodologies across various domains presents a formidable challenge. This had led to an absence of a cohesive and efficacious learning framework within the contemporary SafeRL research milieu. In this work, we introduce a foundational framework designed to expedite SafeRL research endeavors. Our comprehensive framework encompasses an array of algorithms spanning different RL domains and places heavy emphasis on safety elements. Our efforts are to make the SafeRL-related research process more streamlined and efficient, therefore facilitating further research in AI safety. Our project is released at: `https://github.com/PKU-Alignment/omnisafe`.

Keywords: Safe Reinforcement Learning, Learning Framework, Paralleled Acceleration

1. Introduction

Reinforcement Learning(RL) has gained immense attention as a powerful class of machine learning algorithms capable of addressing complex problems in diverse domains such as robotics and game playing (Silver et al., 2016, 2017; Vinyals et al., 2019). Nevertheless, the application of RL in safety-critical domains raises concerns about unintended consequences

∗. Jiaming Ji, Jiayi Zhou, and Borong Zhang are core developers. They contributed equally to this paper.
†. Yaodong Yang is the corresponding author.

and potential harms (Bi et al., 2021). The self-governed learning of RL agents, relying on environmental feedback, can give rise to unsafe control policies. This may result in adverse outcomes, such as autonomous vehicles valuing speed over safety. These concerns have prompted extensive research into developing SafeRL algorithms (Garcıa and Fernández, 2015; Gu et al., 2022) for safety-critical applications (Feng et al., 2023; Vidgen et al., 2024).

Lack of OSS Infrastructure for SafeRL Research Despite the existence of open-source software (OSS) and frameworks, a unified code framework remains elusive in the SafeRL field. OpenAI introduced the safety-starter-agent (Ray et al., 2019), a SafeRL framework in 2019 that implemented four classical algorithms based on TensorFlow-v1. While this framework has aided subsequent researchers, it has not been updated or maintained since its release, and its core dependency, TensorFlow-v1, has been deprecated for years. SafeRL algorithm implementation is complex in nature such that the cross-domain integration with RL algorithms (*i.e.* constraint optimization, safe control theory) poses a significant technical challenge, resulting in the ongoing absence of a unified and effective learning framework within the contemporary SafeRL research landscape. This lack of a unified backend and extensible framework substantially impedes progress in the domain.

To fulfill this gap, we present `OmniSafe` , an infrastructural framework designed to accelerate SafeRL research by providing a wide range of algorithms, including Off-Policy, On-Policy, Model-based, and Offline algorithms, *etc.*

2. Features of `OmniSafe`

The `OmniSafe` [1] framework is featured with the following contributions:

(1) High scalability. `OmniSafe` is a highly scalable framework, employing two distinctive design components: `Adapter` and `Wrapper`. The `Adapter` provides a unified interface for environment interaction to the algorithm, shielding the learning process from variations in different interaction mechanisms (*e.g.,* offline). The `Wrapper` offers plug-and-play features for the environment interaction layer (*e.g.,* action scaling). These abstraction layers enable researchers to reuse all implementations of the environment interaction layer when focusing on a specific part, such as the algorithm implementation (Altman, 1999; Sun et al., 2021; Sootla et al., 2022) or environment customization (Li et al., 2022; Ji et al., 2023).

(2) High-performance parallel computing acceleration. By harnessing the capabilities of `torch.distributed`, `OmniSafe` accelerates the learning process of algorithms with process parallelism. This enables `OmniSafe` not only to support environment-level asynchronous parallelism but also incorporates agent asynchronous learning. This methodology bolsters training stability and expedites the training process via the deployment of a parallel exploration mechanism. The integration of agent asynchronous learning in `OmniSafe` underscores its commitment to providing a versatile platform for advancing SafeRL research.

(3) Code reliability and reproducibility. The algorithms implemented `OmniSafe` have been rigorously tested in Safety-Gym (Ray et al., 2019) and Mujoco-Velocity (Zhang et al., 2020) environments to confirm their consistency with the results presented in the original papers. Code examples necessary for result reproduction are provided within the project.

1. More detailed experimental results can be referred to online documentation: `https://www.omnisafe.ai`.

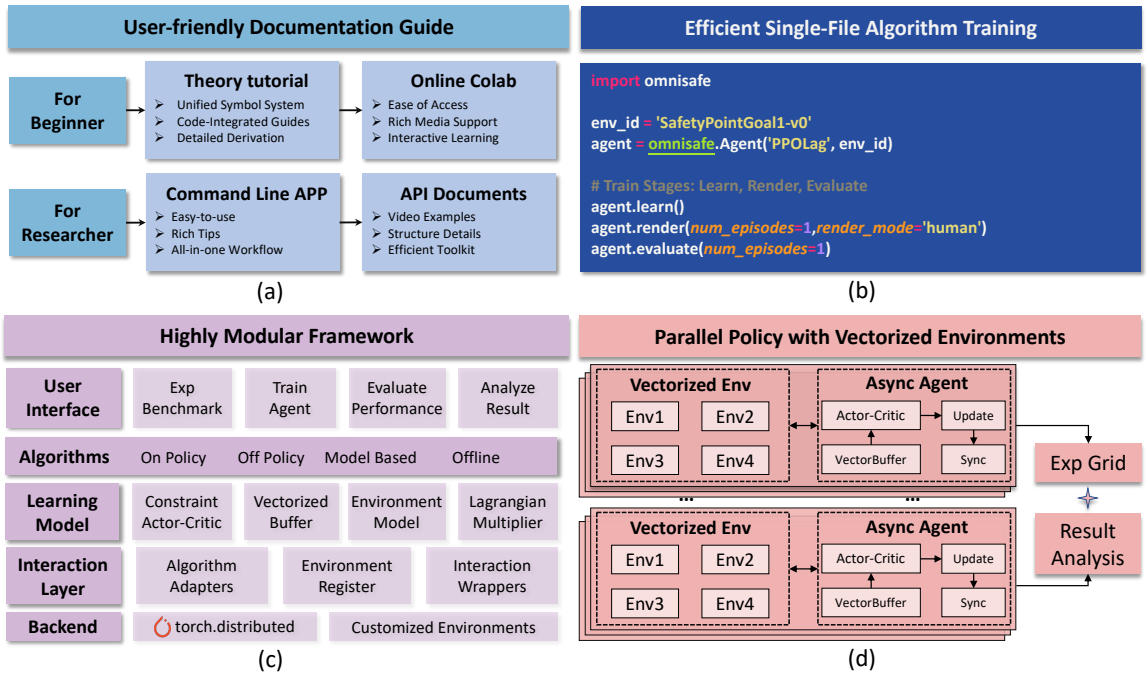

Figure 1: The core features of `OmniSafe` include (a) Comprehensive API documentation with user guides, examples, and best practices for efficient learning, can be found in online documentation; (b) Streamlined algorithm training through single-file execution, simplifying setup and management. (c) Achieve versatility through the utilization of algorithm-level abstraction and API interfaces; (d) Enhanced training stability and speed with environment-level asynchronous parallelism and agent asynchronous learning.

Furthermore, observations, analyses, and raw data from this process will be made accessible to the community as a valuable reference. To maintain stringent code quality, `OmniSafe` 's code has undergone thorough unit testing throughout the development process and adheres to project management standards such as `GitHub Actions CI`, `Pylint`, and `MyPy`.

(4) Fostering the Growth of SafeRL Community. `OmniSafe` is a unified learning framework that represents a significant advance in standardizing the field of SafeRL. The platform offers comprehensive API documentation, including user guides describing both fundamental and advanced features, illustrative examples, and best practices for algorithm parameter selection and result monitoring. Moreover, the documentation delves into detailed theoretical algorithm derivations alongside practical tutorials, thereby facilitating a swift grasp of the SafeRL algorithm for novices.

3. DataFlows of `OmniSafe`

To accommodate the rapid development of various algorithms and mitigate incompatibilities across different environment interfaces, we have devised a unified dataflow interaction framework for `OmniSafe` as illustrated in 2. The utilization of `Adapter` and `Wrapper` designs in `OmniSafe` facilitates enhanced reusability and compatibility with current code resources,

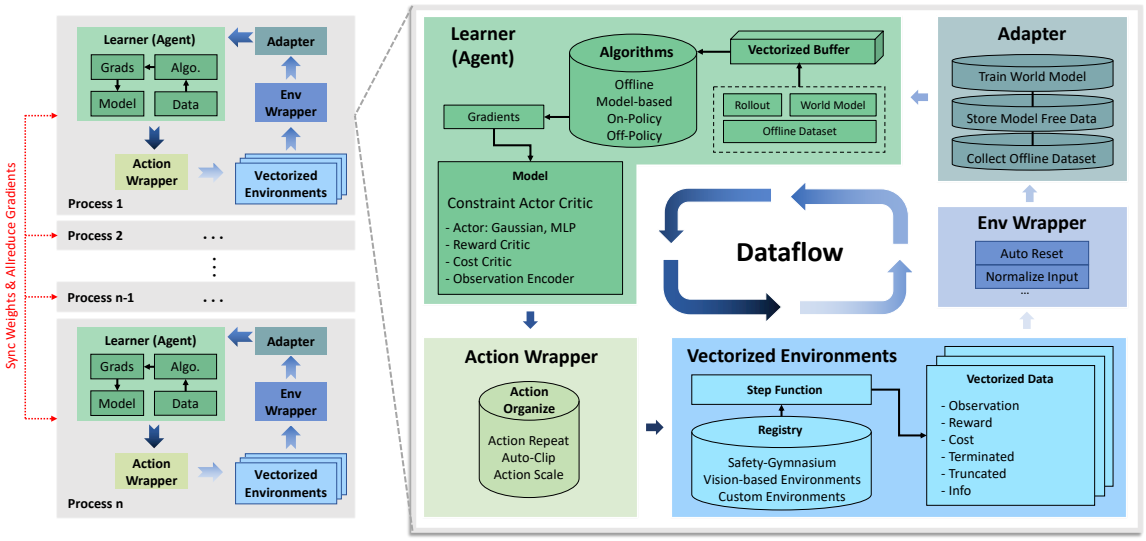

Figure 2: A high-level depiction of `OmniSafe` 's distributed dataflow process. Each process periodically syncs weights and all-reduce gradients with other processes. `Vectorized Environments` first generate trajectories of the agent's interactions with the environment. Second, the `EnvWrapper` monitors and governs the environment's status (*e.g.* Auto-Reset) and outputs. Then, the `Adapter` assigns a suitable execution plan that handles data preprocessing. Next, the `Learner` gathers pre-processed data, calls the learning algorithm, and trains the model. Lastly, the `ActionWrapper` transforms the model's outputs to the agent's actions interpretable by the environments. Thereby completing a cycle of dataflow.

thereby reducing the need for additional engineering efforts when creating new algorithms or integrating new environments. A key highlight is that the `Adapter` layer also provides a convenient approach for transforming problem paradigms and extracting information from the data inflow. For instance, to convert the CMDP (Altman, 1999) paradigm to the SauteMDP (Sootla et al., 2022) paradigm, one simply needs to replace the `Adapter` on the inflow side of data with that of the SauteMDP paradigm. This approach enables users to concentrate more on the problem at hand while incurring minimal code alterations by utilizing the modular tools provided by `OmniSafe` .

4. Conclusion and Outlook

In conclusion, `OmniSafe` provides a comprehensive infrastructural framework designed to expedite SafeRL research, featuring extensive API documentation, compatibility with popular environments, high modularity, and support for distributive computing. We have experimentally validated the implementation of numerous algorithms spanning diverse RL domains, facilitating efficient experimentation and verification of novel concepts. Our documentation delivers practical advice for conducting SafeRL research and supplies a developer guide for enhancing the platform's functionality. Our endeavors are directed toward the standardization of SafeRL-related research tools and methods, with the aim of cultivating a more streamlined and highly efficient approach. By doing so, we endeavor to facilitate and expedite further scientific exploration in the field of AI safety.

Acknowledgments

This work is sponsored by National Natural Science Foundation of China (62376013), Beijing Municipal Science & Technology Commission (Z231100007423015), Young Elite Scientists Sponsorship Program by CAST (2022QNRC003).

References

Eitan Altman. *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.

Zhu Ming Bi, Chaomin Luo, Zhonghua Miao, Bing Zhang, WJ Zhang, and Lihui Wang. Safety assurance mechanisms of collaborative robotic systems in manufacturing. *Robotics and Computer-Integrated Manufacturing*, 67:102022, 2021.

Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953):620–627, 2023.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications. *arXiv preprint arXiv:2205.10330*, 2022.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7:1, 2019.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

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

Aivar Sootla, Alexander I Cowen-Rivers, Taher Jafferjee, Ziyan Wang, David H Mguni, Jun Wang, and Haitham Ammar. Sauté rl: Almost surely safe reinforcement learning using

state augmentation. In *International Conference on Machine Learning*, pages 20423–20443. PMLR, 2022.

Hao Sun, Ziping Xu, Meng Fang, Zhenghao Peng, Jiadong Guo, Bo Dai, and Bolei Zhou. Safe exploration by solving early terminated mdp. *arXiv preprint arXiv:2107.04200*, 2021.

Bertie Vidgen, Adarsh Agrawal, Ahmed M Ahmed, Victor Akinwande, Namir Al-Nuaimi, Najla Alfaraj, Elie Alhajjar, Lora Aroyo, Trupti Bavalatti, Borhane Blili-Hamelin, et al. Introducing v0. 5 of the ai safety benchmark from mlcommons. *arXiv preprint arXiv:2404.12241*, 2024.

Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, et al. Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, 2, 2019.

Yiming Zhang, Quan Vuong, and Keith Ross. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33:15338–15349, 2020.