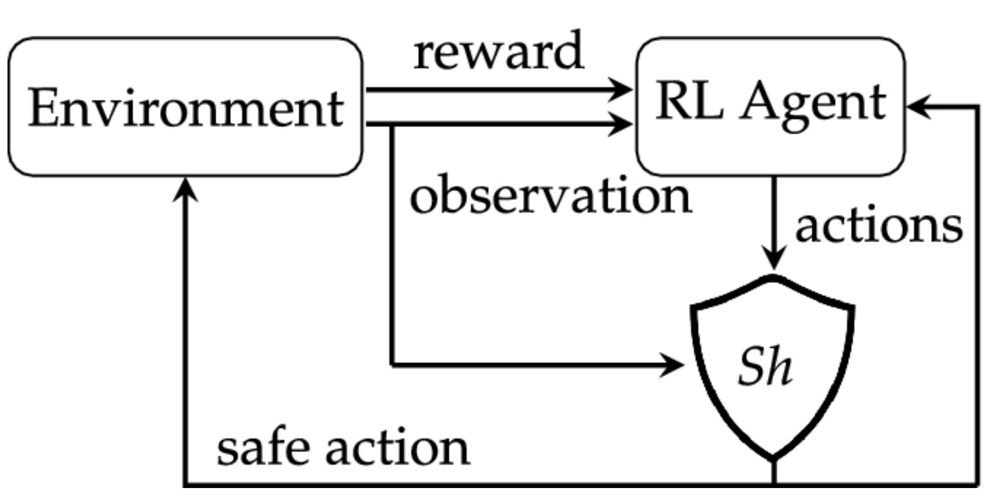
Verification-Guided Shielding for Deep Reinforcement Learning

Davide Corsi[1], Guy Amir[2], Andoni Rodríguez[3,4], César Sánchez[3], Guy Katz[2], and Roy Fox[1]

[1]University of California: Irvine, USA
[2]The Hebrew University of Jerusalem, Israel
[3]IMDEA Software Institute, Spain
[4]Universidad Politécnica de Madrid, Spain

Despite their successes, DRL-based policies often suffer from poor reliability on specific corner cases and unexpected input configurations, which limits their use in safety-critical domains. As a case study, we apply our approach to a real-world robot navigation problem combining the strenghts of **shielding** and **verification of DNNs**.

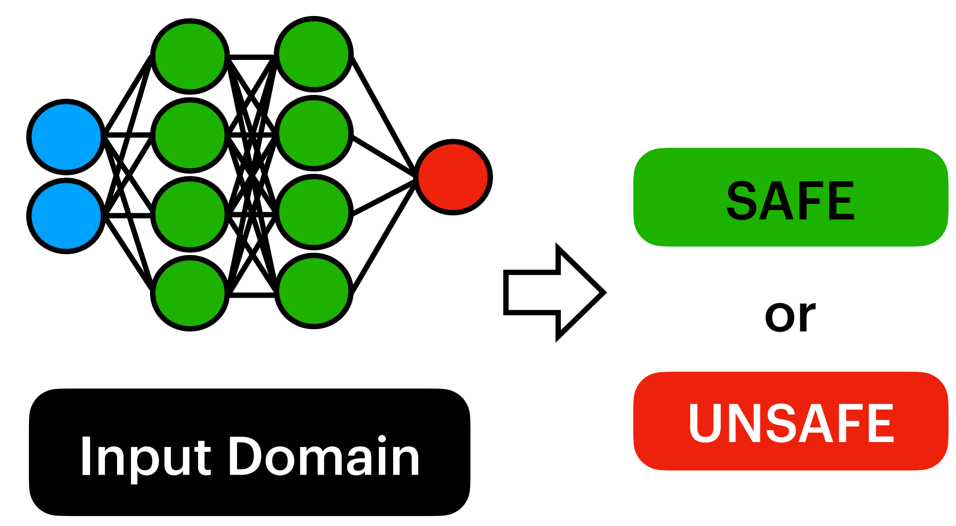
Shielding in Reinforcement Learning



A shield is an external component that can certify every action selected by the agent to guarantee the safety.

Calling an external nonlinear solver at each time step is **computationally extremely expensive**, preventing a real time execution.

Verification of Neural Network

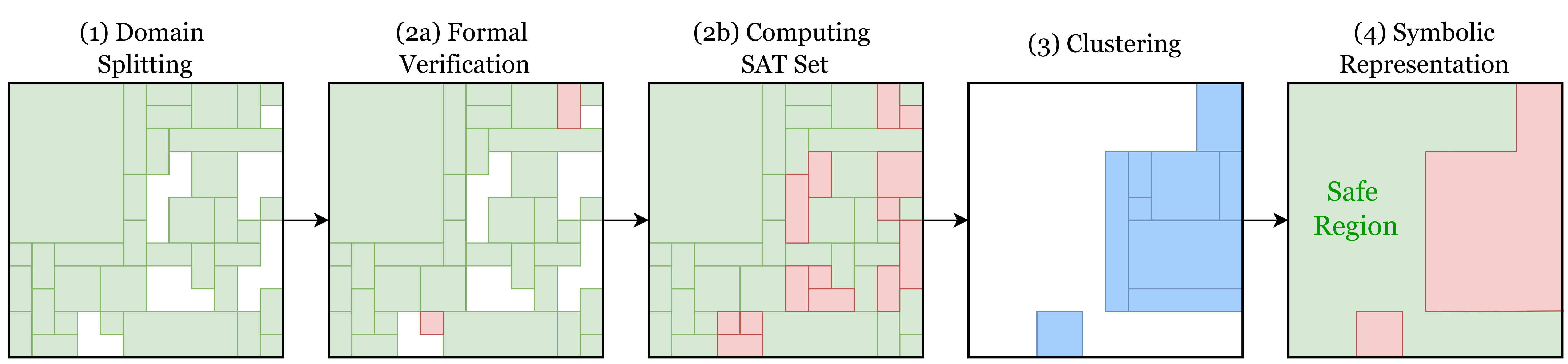


Given a DNN and a set of requirements on the input domain, a verification tool return a binary answer.

It is unlikely for a neural network to be completely safe for any input, and once declared *UNSAFE, it* cannot be easily fixed.

Verification-Guided Shielding

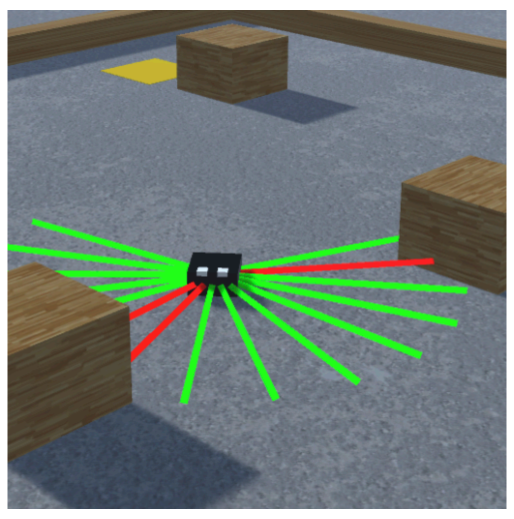
GOAL
Minimizing the calls to the shield [3] while preserving the safety gurantees

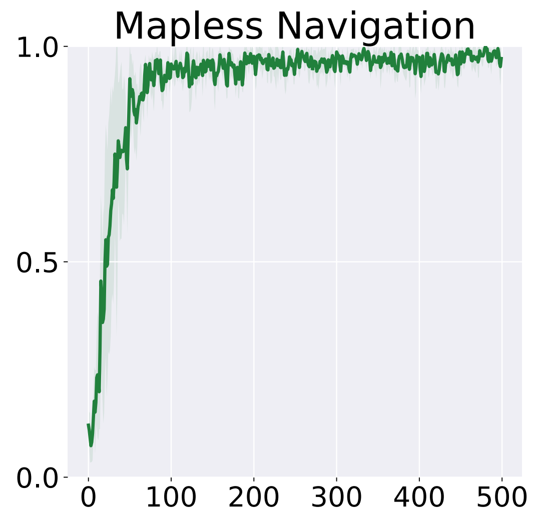


- Split the input domain into potentially safe regions [2] before a formal verification step on the generated subdomains [1].
- Generation of a provable safe set where the shield is not needed, while the agent is potentially unsafe elsewhere.
- Clustering and Symbolic Representation step to reduce the complexity of the online checking process.

[1] **Formal Verification of Neural Networks for Safety-Critical Tasks in Deep Reinforcement Learning.** D. Corsi, E. Marchesini et al.; UAI 2021.
[2] **The #DNN-Verification problem: Counting Unsafe Inputs for Deep Neural Networks.** L. Marzari, D. Corsi et al.; IJCAI *2023.*
[3] **Shield Synthesis for LTL Modulo Theories.** A. Rodriguez, G. Amir, D. Corsi et al.; arXiv *2024.*

Experimental Results





Seed	Full Shield		Verification-Guided Shield		Gain (%)
	Active Time (%)	Overhead	Active Time (%)	Overhead	
12	100	40.0×	28.6	14.1×	64.8
66	100	32.5×	32.4	13.1×	59.7
239	100	36.3×	44.5	21.5×	40.7
251	100	31.1×	37.6	13.2×	57.6
258	100	35.5×	33.8	13.9×	60.1
104	100	4.8×	61.7	3.6×	25.1
225	100	4.4×	53.1	3.5×	20.5
239	100	4.5×	2.1	1.8×	60.0
243	100	4.5×	1.3	1.6×	71.1
310	100	4.6×	3.4	1.5×	67.4

This table highlights the advantage of using our approach, we **drastically reduce the number of calls to the solver**, increasing the performance of the agent towards a real-time execution while preserving the safety guarantees.

Future Directions

➡Learn the shield during the training loop *(eliminating the need to keep it enabled at execution time)*.
➡A novel solution to prove wether a shield can *always* return a valid and safe action.
➡An automatic approach to design safety requirements.

?

Contact author: dcorsi@uci.edu

UCIRVINE

Check out the full paper for more details and proof!