Formal Analysis of Deep Decision-Making Models for Aquatic Navigation

Davide Corsi
Department of Computer Science
University of Verona
Verona, Italy
davide.corsi@univr.it

Enrico Marchesini
Department of Computer Science
University of Verona
Verona, Italy
enrico.marchesini@univr.it

Alessandro Farinelli
Department of Computer Science
University of Verona
Verona, Italy
alessandro.farinelli@univr.it

***Abstract*—We consider the problem of aquatic navigation, using formal analysis to verify decision-making models trained with Deep Reinforcement Learning (DRL). This learning task is difficult due to the non-stationary environment, and the uncertainties of the robotic hardware. For these reasons, it is crucial to develop safe and efficient training solutions, ensuring the correct behavior of the network to avoid dangerous situations (e.g., platform accidents). We address the safety of the resulting models by using a novel formal verification approach for decision-making problems, based on interval analysis. We show that our verifier allows to provably guarantee whether the trained models respect a set of desired safety properties, reducing the computational requirements of previous frameworks. Crucially, in our experiments, the low number of property violations allows to design a controller that can avoid undesirable behaviors and hence guarantee safety.**

***Index Terms*—Reinforcement Learning, Robotics, Safety**

I. Introduction

Successful applications of Deep Reinforcement Learning techniques in real scenarios are driven by the presence of physically realistic simulation environments [1]. In this paper, we consider the well-known DRL task of robotic navigation to introduce a novel aquatic navigation problem, characterized by a physically realistic water surface with dynamic waves. In detail, we consider the drones of the EU-funded Horizon 2020 project INTCATCH (https://www.intcatch.eu/) as robotic platform. Among the variety of applications for aquatic drones, autonomous water quality monitoring represents an cost-effective alternative to the more traditional manual sampling [2]. To this end, we aim at combining the navigation task and these interesting robotic platforms, to train a robust policy that can autonomously navigate to random targets, avoiding collisions. Our novel aquatic scenario is therefore an important asset to benchmark and tests the autonomous capabilities of aquatic drones. Recent applications of autonomous robotic navigation are typically addressed by DRL, due to its ability to train decision-making policies and adapting to previously unknown environments [3]. These solutions, however, present several challenges that prevent a wider utilization of DRL. First, DRL has to cope with the environment uncertainties, requiring a huge number of trials to achieve good performance in the training phase. In this learning paradigm, the lack of diverse exploration, when operating in high-dimensional spaces (such as our dynamic water scenario), also causes convergence to local optima. We address these issues within an emergent research direction that proposes the combination of gradient-free and gradient-based DRL, to assimilate the best of both solutions [4]. The idea behind our approach is to incorporate the sampling efficiency of value-based DRL while diversifying the collected experiences using the gradient-free population.

Robotic tasks usually involve high-cost hardware, hence the behavior of the trained policy must be evaluated to avoid undesirable and potentially dangerous situations. We address the analysis of the behavior of these models, by extending previous formal verification approaches [5] to the domain of decision-making. In detail, models that are trained for decision-making, require the analysis of multiple outputs (e.g., choose the action that maximizes a return), and prior works does not consider such relations. In contrast, they aim at directly verifying whether the bound of a single network output lies in a given interval (e.g., a motor velocity never exceeds fixed bounds). Hence, our method uses interval analysis [6] to verify the relations between two or more network outputs (e.g., in a certain input domain, the network must not select a certain action). This enables the verification of DRL models, where the output nodes correspond to actions that are selected by the trained network, according to their value.

II. Formal Verification

State-of-the-art approaches rely on Moore's interval algebra [7] to compute strict bounds for the values that each output of the DNN assumes. In detail, these methods propagate a subset of the input domain layer by layer and apply non-linear transformations (when required), to compute the corresponding interval for each output node of the network. Hence, they verify if it lies in the desired range. We refer to the input and output intervals as *input area* and *output bound*, respectively.

A. Decision-Making Tasks

Our combined value-based approach encodes the navigation task a decision making problem. In these contexts, the agent typically selects the action to perform that corresponds to the node with the highest value. For this reason, in contrast to the previous safety properties, we propose a different formulation that is specifically designed for our class of problems:

$$\Theta : \text{If } x_0 \in [a_0, b_0] \wedge ... \wedge x_n \in [a_n, b_n] \Rightarrow y_j > y_i \qquad (1)$$

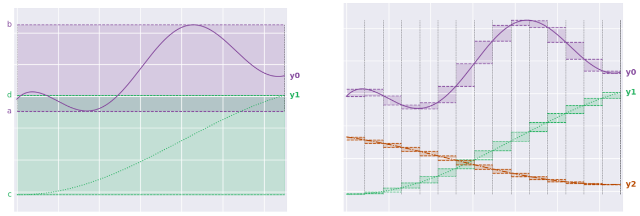
We refer to these properties as *safe decision properties* as they are used to ensure that a given action (e.g., y_j) is always preferred over the others for a given input configuration. To verify if an interval $y' = [a, b]$ is greater than another one $y'' = [c, d]$, we rely on the interval algebra. In particular, on the preposition:

$$b < c \Rightarrow y' < y'' \qquad (2)$$

Figure 1a shows the limitations of prior methods in the verification of decision-making tasks, where y_0 and y_1 represent the output functions generated by two nodes of a generic network. In detail, it is not possible to infer which output action will be selected as they only compute the output bounds, without considering the shape (and the relationship) of the output functions. Since we can not guarantee Prop. 2 we can not assert anything on the specified property. In contrast, the main idea behind our novel method is to perform an estimation of the output curve for each node. Figure 1b summarizes our approach, where by subdividing the initial area and calculating the corresponding output bounds for each interval, we obtain: (i) a sensible reduction for the overestimation problem [8], and (ii) a good estimation of the output curves. Finally, by applying Prop. 2 on each sub area, we can obtain three possible results from the analysis of a safe decision property: (i) the property holds, (ii) the property is violated, or (iii) we can not assert anything on the property in this area. In the third case, we iterate the partition of the interested area to increase the accuracy of the estimation. Moreover, our approach can compute the portion of the starting area (and eventually returns it as a counterexample) that causes the violation of a specific property. By normalizing this value, we obtain a novel informative metrics, the *violation rate*.

B. Safety Properties

The design of a set of safety properties for our aquatic navigation scenario, that presents a variable set of obstacles, is a challenging problem. Given the dynamic and the non-stationary nature of such environment, it is not possible to formally guarantee the safety of the drone in any possible situation. For this reason, we focus on ensuring that the agent always makes rational decisions (i.e., it selects the best action given available information). Consequently, we selected three properties, that represent a possible safe behavior of our agent in relation to the presence of obstacles:



(a) One output function with one subdivision.

(b) Multiple output curves with a subdivided input area.

Fig. 1: Explanatory output analysis of our formal verification approach.

	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
Θ_0	9.3 ±2.4	5.3 ±3.1	0.0 ±0.0	1.3 ±0.3	1.3 ±0.7
Θ_1	3.0 ±2.3	4.1 ±2.2	0.0 ±0.0	0.2 ±0.2	0.0 ±0.0
Θ_2	7.1 ±1.4	6.9 ±2.7	3.4 ±0.2	3.6 ±1.4	6.3 ±2.6

TABLE I: For each property we show the mean and the variance of the violation rate (%) of the best 15 models for each seed (considering the success rate).

Θ_0: If there is an obstacle near to the left, whatever the target is, go straight or turn right.

Θ_1: If there is an obstacle near to the right, whatever the target is, go straight or turn left.

Θ_2: If there are obstacles near both to the left and to the right, whatever the target is, go straight.

III. Results

Table I shows the results for each property, considering the violation rate over the five different seeds used in our training. Results show that the violation rate and the success rate are not necessarily related; despite we tested the safety on the best models, we found in some cases a high violation rate. For this reason, our safety analysis is a necessary step to evaluate a policy before its deployment in a real-world environment. Finally, due to the low violation rate of our best model, it is possible to design a simple controller to guarantee the correct behavior of the network. The decision-making frequency of the robot is set to $10Hz$ and, with the violation rate presented in Table I, a complete search through the array of the sub-areas that cause a violation, always requires less than 0.09s. This means that, with our hardware setup, we can verify if the input state leads to a violation at each iteration, without lags in the robot operations. Consequently we can avoid all the decisions, derived from input configurations, that lead to the violation of our desired safety properties. The video attached describes in more detail the simulation environment and provides some visual examples of the behaviours executed by our DRL method.

References

[1] A. Ray, J. Achiam, and D. Amodei, "Benchmarking Safe Exploration in Deep Reinforcement Learning," in *OpenAI*, 2019.

[2] A. Castellini, D. Bloisi, J. Blum, F. Masillo, and A. Farinelli, "Multivariate sensor signals collected by aquatic drones involved in water monitoring: A complete dataset," in *Data in Brief*, 2020.

[3] E. Marchesini and A. Farinelli, "Discrete deep reinforcement learning for mapless navigation," in *ICRA*, 2020.

[4] ——, "Genetic deep reinforcement learning for mapless navigation," in *AAMAS*, 2020.

[5] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Efficient formal safety analysis of neural networks," in *NIPS*, 2018.

[6] D. Corsi, E. Marchesini, and A. Farinelli, "Formal verification for safe deep reinforcement learning in trajectory generation," in *IRC*, 2020.

[7] R. E. Moore, "Interval arithmetic and automatic error analysis in digital computing," Ph.D. dissertation, Stanford University, 1963.

[8] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *USENIX Security Symposium*, 2018.