Safety first: Toward safe action selection with contextual affordances

Inés Apablaza[1], Martín Saavedra[1], Angel Ayala[2], Bruno Fernandes[2], and Francisco Cruz[1,3]
[1]Escuela de Ingeniería, Universidad Central de Chile, Santiago, Chile
[2]Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil
[3]School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
Emails: {ines.apablaza, martin.saavedra}@alumnos.ucentral.cl, {aaam, bjtf}@ecomp.poli.br, f.cruz@unsw.edu.au

***Abstract*—Reinforcement Learning empowers agents to make decisions in environments with the aim of maximizing a defined reward. In the area of robotic exploration, the challenge lies in enabling the agent to navigate its surroundings while avoiding dangerous states. Contextual affordance, a predictive model, anticipates the consequences of actions based on the agent's current state. This model proves invaluable in guiding the agent away from dangerous situations. In the pursuit of enhancing robotic exploration safety, this paper examines the efficacy of a robotic agent employing contextual affordance to steer clear of unsafe states during exploration. The evaluation encompasses both a controlled environment and an uncontrolled setting. The results obtained show the learning agent is able to avoid dangerous states more effectively when using contextual affordance. By contrasting the agent's performance in these scenarios, our study reveals a pronounced improvement in the agent's ability to avoid unsafe states when leveraging contextual affordance.**

***Index Terms*—safety robotics, contextual affordances, reinforcement learning**

I. Introduction

Reinforcement learning (RL) [1] is a crucial approach in developing autonomous, intelligent robotic agents because it can learn from experience. This approach imitates biological learning systems, mirroring how animals learn by interacting with their environment. This interactive process generates a data flow, enabling the agent to determine and adapt the most effective actions to achieve its goal. The learning process through trial and error requires an agent capable of evaluating its current state to apply an action, expecting a high reward value. After a determined number of interactions, the agent must be able to choose the best action, exploiting its knowledge about the environment. Nevertheless, for that to happen, the agent should have explored different actions for a given state to discover new outcomes and make a relationship from that. In this regard, exploring possible better actions may lead to undesired states, which can damage the robotic agent or cause it to enter into an anticipated final state without completing the learning process.

The robotic agent's associated risk under RL action exploration refers to the probability of entering into a catastrophic or unwanted state [2]. This situation prevents the agent from completing its task or solving a problem inefficiently [3]. In this regard, safe exploration seeks to improve RL action exploration while preventing the agent from entering a dangerous state.

Contextual affordance (CA) is an effective, safe robotic model that anticipates the effects of an action performed by an agent [4]. CA models allow the agent's action effect prediction by knowing its state, the object that wants to interact, and the action to be performed. In most cases, this action effect is modeled by an artificial neural network that receives the agent's state, object, and action information. In the RL framework, CA works as an action guide by estimating its effects on processing the current robot state. Nevertheless, one main problem is that the robotic agent must explore all the different states and actions to optimize the action effect model. Consequently, the robot is forced to iterate at least once through these potentially dangerous or undesirable scenarios.

As an instance of developmental learning, controlled scenarios in real-life settings, particularly during a child's early learning stages, have proven successful in creating valuable experiences [5]. These scenarios effectively expose the child to the potential dangers associated with various actions. For instance, when a child experiences a controlled fall within a crib, they learn about the risks linked to falling. This controlled setting allows the child to grasp the dangers without actually encountering them.

Similar to this approach, in this work a controlled setup is introduced for the RL agent to learn the action effect model and identify unsafe states of the environment.

The controlled setups are constructed manually to replicate key aspects of the original scenario. This involves maintaining the same number of actions and objectives as the original scenario, with the primary goal of teaching the agent to avoid dangerous states while still completing its tasks. Additionally, the controlled setups feature the same types of dangerous states as those present in the original scenario. For instance, it would be counterproductive to train for a type of dangerous state that does not exist in the original setup. The proposed controlled scenarios are scaled-down versions of the original scenario, typically reducing in size by 60% to 70%, while preserving the underlying logic of the original scenario.

The purpose of utilizing these controlled setup is to generate sufficient data to train the CA model before deploying the agent to an uncontrolled setup. By exposing the agent to scenarios with controlled variations, we can effectively train the model to recognize and respond to various dangerous states with the same underlying logic as those encountered in the

979-8-3503-7457-5/24/$31.00 ©2024 IEEE

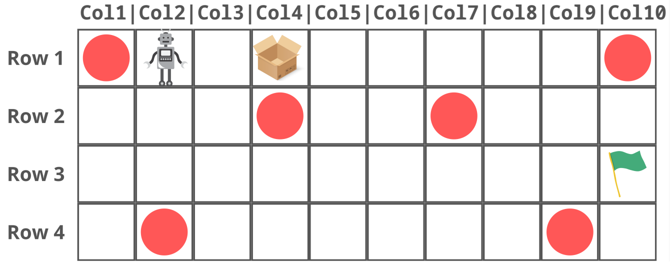

Col1 | Col2 | Col3 | Col4 | Col5 | Col6 | Col7 | Col8 | Col9 | Col10
Row 1
Row 2
Row 3
Row 4

Fig. 1. An example grid world scenario is composed of four rows and ten columns. The robotic agent from its initial position must pick up the box and move it to reach the green flag's goal position as fast as possible. Inside the scenario, there are six red circles representing unsafe states or positions the agent must avoid. In this environment, contextual affordances become very handy to prevent the agent from reaching those unsafe states.

original scenario.

Through this joint learning using CA, it is expected that by transferring the agent from the controlled scenario to the original or uncontrolled setup, the agent will be able to explore actions without entering unwanted or dangerous states of the environment, thus enabling safe exploration.

II. Related Background

A. Contextual affordance

This probabilistic model was inspired by Gibson's psychological study of the Affordance concept [6], showing satisfactory results in robotics [7]. Affordance establishes a relationship between the agent's actions and the objects in the environment, or the agent's state generally speaking, considering that an object might be part of its current state. This relationship is a tuple composed of three elements $<\text{object}, \text{action}, \text{effect}>$, from which it is possible to discover one element knowing the other two. Therefore, knowing the agent's action and the object with which it is going to interact, it is possible to infer the effect of that interaction [8] by solving Eq. (1):

$$\text{effect} = f(\text{object}, \text{action}). \tag{1}$$

When applying affordances in different reinforcement learning scenarios, there is a particular situation in which the effects vary depending on the robot's current state. For example, consider the scenario in Fig. 1. In this scenario, the robot must pick up the box and move it to reach the green flag's position with the least number of actions. However, there are unsafe locations or undesirable states represented by red circles that the agent must avoid.

In order to reach the goal position, the agent will begin exploring different actions to learn how to behave to move the box to the green flag as fast as possible. Since there are unsafe positions, the use of CA will improve the decision-making ability of the agent to avoid such states. This is achieved through effect prediction for a given action that will or will not fall into those undesirable states.

Considering the illustrative scenario in Fig. 1 and if we assume the agent has already picked up the box in position $(row_1, column_4)$, then the agent might choose a downward movement with the box. According to Eq. (1), an action effect is obtained such that $\mathit{effect}_1 = f(box, down)$ and a new state $(row_2, column_4)$ is reached. As the new agent's state is an undesired state, the effect_1 can be associated with an unsafe transition. Nevertheless, considering the same transition (i.e., downward movement with the box) from a new initial state such as $(row_1, column_6)$, a different action effect is obtained $\mathit{effect}_2 = f(box, down)$ and a new state $(row_2, column_6)$ is reached. Since the new state is not dangerous, the action effect effect_2 can be associated with a safe state transition.

By analyzing both equations obtained and performing the same action, it can be concluded that $\mathit{effect}_1 \neq \mathit{effect}_2$, whereas in the state $(row_1, column_2)$ and the state $(row_1, column_6)$ the $left$ action leads to an unsafe and safe transition, respectively. In this situation, the inclusion of the robot's current state is proposed [9] to differentiate these effects $f(state_1, box, down) \neq f(state_2, box, down)$, obtaining the CA equation formalized as:

$$\text{effect} = f(\text{state}, \text{object}, \text{action}). \tag{2}$$

B. Reinforcement Learning

The SARSA on-policy learning algorithm was chosen as the RL framework to follow. It comprises a tuple of current state and action, with the obtained reward value, and the next state and action as follows $< s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1} >$, which originated the name. The choice of the algorithm is based on the update of the Q values, which is based on the reward and the immediate next action as formalized in (3). SARSA is considered an online policy optimization by taking into account the immediate reward and the long-term effects. Additionally, its behaviour of choosing an action from the current state and immediately choosing the following action for the next state increases the probability of falling into an undesired state [10].

$$\begin{aligned} Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} \\ + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \Big]. \end{aligned} \tag{3}$$

An important factor in RL is the tradeoff between exploration and exploitation of the agent action selection to discover potential outcomes from iterations with the environment. In this regard, the ϵ-greedy technique is used where ϵ indicates the probability of choosing a random action for exploration. On the contrary, the probability of exploiting the agent's knowledge to choose an action is given by $1 - \epsilon$. Hence, the higher the ϵ value, the greater the probability of selecting a random action, prioritizing exploration instead of exploitation [1].

During training, it is expected that the agent explores more at the beginning and exploits more at the end. In this work, the technique of decaying ϵ was also considered, where the ϵ starts with the value of 1 and decreases proportionally until reaching a minimum value of 0.01. This will allow the agent to prioritize exhaustive action exploration in both controlled and uncontrolled setups to finally prioritize actions with a higher Q-value.

III. Experimental Setup

A. Contextual Affordance Integration

The robotic agent's data acquisition and contextual affordance model implementation process unfolds two models in the research context. The first is obtained from the agent's experiences within the controlled setup under extensive exploration. Following the ϵ-greedy strategy with an ϵ value of 1, the agent will go through a broad spectrum of state and action pairs. This approach is instrumental in encouraging the agent to explore comprehensively. The obtained agent's data is used for training a multilayer perceptron, as the contextual affordance model definition. The training process followed a supervised approach where the network input data was the action and the state. In this training phase, the data is shuffled randomly before each epoch to prevent bias and improve model generalization. The multilayer perceptron was employed with logistic activation and up to 5,000 iterations. This quantity allows the algorithm to thoroughly explore the parameter space and refine the model weights. We experimented with fewer iterations, but observed diminished performance during training. The output of the CA model ranges between 0 and 1, where the threshold defines the value it will take, with 0 for normal cases and 1 for unsafe states. The second is the agent model, trained in an extension of the RL framework, presented in Figure 2. The robotic agent is transitioned to the regular setup once the CA model is trained in the controlled setup. For each movement executed, the agent delivers the current state and action taken to the CA model. The CA model infers an effect percentage value, indicating how dangerous a state is. A threshold value of 70% indicates if the current state is dangerous or not. This threshold value was chosen through analysis and experimentation to strike an appropriate balance between sensitivity and specificity in classifying states as safe or unsafe. Setting a threshold too low might lead to an overestimation of unsafe states, resulting in overly cautious behavior by the agent and reduced efficiency in completing its task. Conversely, setting it too high might lead to an underestimation of danger, exposing the agent to unnecessary risks. The 70% threshold was determined as a suitable compromise between these extremes, it offers a reasonably high level of safety in identifying potentially unsafe states while allowing the agent to maintain its efficiency in the task. If the predicted effect surpasses this threshold, the agent changes its action, opting for an alternative. This iterative process continues, ensuring that the agent constantly refines its actions based on the feedback from the CA model.

It implies that the effect prediction represents confidence; when exceeding the threshold, the agent acknowledges a substantial risk of encountering a dangerous state, prompting immediate corrective action. In both scenarios, in any state the agent only has one available action or no actions that could potentially lead it to a dangerous state. This implies that whenever the agent considers taking an action that could lead to a danger, it assumes that any alternative action will be inherently ”safe”. Therefore, the immediate corrective action will be any action that promises more reward for the agent except the action considered dangerous, as shown in Figure 2.

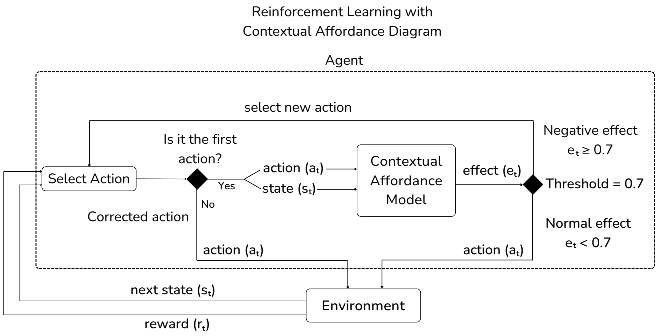


Fig. 2. Reinforcement learning framework with contextual affordance model.

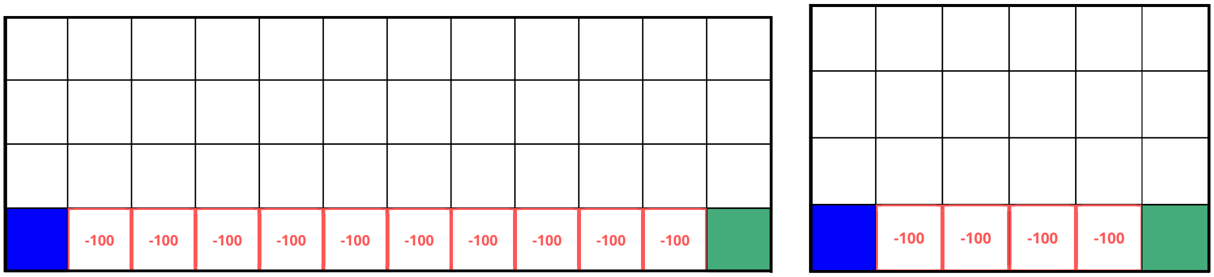
B. Cliff Walking

The Cliff Walking scenario, shown in Fig. 3(a), is a grid world where the agent must reach the goal position in the least number of moves. The complexity of the scenario lies in the agent avoiding the cliff located at the bottom row, which will penalize the agent and send it back to its initial position. The state space in a grid world is ruled by the 2-axis coordinate $(row, column)$, representing each grid position. The initial position is represented by $(0, 4)$ and the goal position is represented by $(12, 4)$. The available actions correspond to four directions, namely, up, down, left, and right, where the first two changes the row number by a negative and positive unit, respectively. The other two actions affect the column number by a positive and negative value.

The agent aims to trace an efficient route to reach the goal position. A possible route can be by the top row, far from the cliff zone, which will be the safest route. Another possibility is at the immediate upper row from the cliff zone, which is not the safest but the shortest path to the goal. For this last case, the agent can easily get into the cliff if, under exploration, the down action is selected.

1) Controlled setup: A reduced grid-world size of 6x4 from the original setup of 12x4 is proposed as the controlled setup. The number of columns reduction decreases from 10 dangerous states to only 4 (Fig. III-B), represented by the cliff zone. This configuration can lead the agent, even choosing the riskiest path, not entering the cliff during action exploration.

2) Neural network architecture: The neural network architecture of the Cliff Walking scenario consists of an input layer with 4 neurons corresponding to [State , Action, X-axis Position, Y-axis Position], a hidden layer with 190 neurons that utilizes a sigmoid activation function, and an output layer with a single neuron also using a sigmoid activation function. This architecture remains consistent throughout both the controlled setup for training and the subsequent uncontrolled setup.



(a) Original setup.

(b) Controlled setup.

Fig. 3. Graphical representation of the Cliff Walking scenario. The blue square corresponds to the robotic agent, the green square to the goal, and the red squares correspond to the cliff.

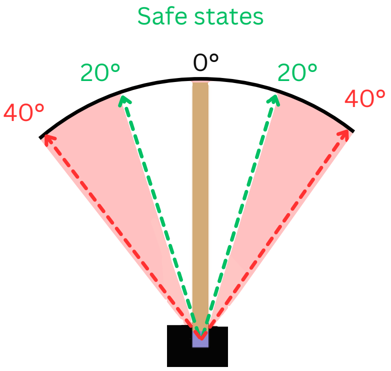


Fig. 4. Controlled Cart Pole setup. Representation of the unsafe states of the controlled scenario. Angles between -20° and 20° represent a safe state for the agent, and beyond that represent unsafe states.

C. Cart Pole

The Cart Pole scenario's goal is to vertically balance a rigid pole attached to a cart for as much time as possible. The agent can move on a horizontal frictionless axis to balance the pole. The state space is represented by a 4-elements vector representing the cart's position and velocity and the pole's angle and angular velocity. The agent can move left or right, applying a constant positive or negative force on the horizontal plane, respectively.

The agent aims to keep the pole perpendicular to the cart's surface by applying the correct movements. Nevertheless, unsafe states prevent the agent from completing the task efficiently. When the pole reaches a rotation angle between -20° and 20°, it will be considered a safe state since, within that rotation, the agent can easily balance the pole (see Fig. 4). When the pole's inclination is beyond 20° and less than -20°, it will be considered an unsafe state since the agent must apply the same movement many times to get back on track.

1) Discretize states: As the state space of the Cart Pole is represented in the continuous domain, the 4-elements vector was discretized in this work. This state discretization aims to reduce the problem's complexity required to find an optimal learning policy. As previously demonstrated [11], the BOXES technique [12] is suitable for discretizing the Cart Pole state, which consists of splitting the continuous state space into some boxes or bins representing an interval from the continuous space. The current continuous value is then assigned to a bin from the corresponding range interval.

TABLE I
RANGE AND NUMBER OF BINS FOR EACH VARIABLE IN THE CART POLE STATE VECTOR.

Variable	Range	Number of bins
Angle of the pole (θ)	[-180° ; 180°]	90
Angular velocity (θ')	[-4 ; 4]	8
Position of the cart (x)	[-4.8 ; 4.8]	1
Velocity of the cart (x')	[-4 ; 4]	1

In the scenario, the position of the cart and its velocity are not considered relevant for the goal of balancing the pole, so for both variables, only one box was defined. In this regard, the discretization only represents the cart on the left or right sides for both position and velocity. On the contrary, for rotation variables of the pole and as they are more significant for the balance, 90 and 8 boxes were assigned for the rotation angle and angular velocity, respectively. The BOXES implementation details are described in Table I. Therefore, only two components from the 4-vector state were considered relevant to predict unsafe states. One corresponds to the pole's angle (θ) and the pole's angular velocity (θ').

2) Controlled setup: The controlled setup, as in Cliff Walking, corresponds to a limited representation of the original setup. In the original setup, the pole can obtain angles from 0° to 360°, while in the controlled setup the range is limited to a circular segment from -40° to 40° (Fig. 4).

For the controlled setup of the Cart Pole, the rotation angle was limited to represent a safe state and an unsafe state from its original setup. In the original setup, the pole can obtain angles from -180° to 180°, representing the pole above the horizontal plane; in the controlled setup, the range was limited to a circular segment from -40° to 40°, as shown in Figure 4. Under that range, the safe state is represented by the rotation angle between -20° and 20°, while if it goes beyond 20° and less than -20° is considered an unsafe state. These angle intervals allow a 50% distribution of unsafe states of the problems.

3) Neural network architecture: Similar to the neural network architecture built for the Cliff Walking scenario, the Cart Pole architecture features an input layer with 4 neurons, a hidden layer with 890 neurons utilizing a sigmoid activation function, and finally an output layer with a single neuron, also using a sigmoid activation function. In this case, the state consists of two variables: one corresponding to the angle of the pole (θ), and the other corresponding to the angular velocity of the pole (θ'). On the other hand, variables such as the position of the cart (x) and the velocity of the cart (x') are not considered relevant in predicting a dangerous state. These are the variables of the input layer: [State (2), Action, Pole Angle Rotation].

IV. Results

A neural network was first trained with the transition data from the controlled setup. Then, the traditional RL-loop was executed for training the agents in the full scenario. A first experiment was done using only RL as a baseline to solve each scenario. A second experiment uses the same scenarios but now with contextual affordance support. The RL algorithm hyper-parameters for both algorithms were empirically determined and are shown in Table II. The training length for each scenario, including episodes and iterations, is shown in Table III.

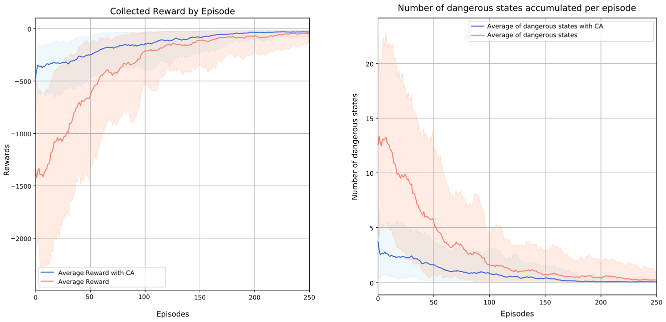
TABLE II
Training parameters used for the agents in both scenarios.

Parameter	**Value**
Discount factor (γ)	0.99
Exploration rate (ϵ)	[1 ; 0.01]
Decay rate of ϵ	0.995
Learning rate (α)	0.1

TABLE III
Number of episodes and iterations per scenario.

Scenario	**Number of episodes**	**Number of iterations**
Cliff Walking	250	100
Cart Pole	500	100

Each agent was trained ten times on each scenario, computing its average, maximum, and minimum accumulative reward for performance comparison. Both agents, the autonomous RL and RL using CA support, solved the scenario problem and performed good learning curves. Nevertheless, using CA highlights the reward value variability reduction against vanilla algorithm implementation. This can be observed by the distance between the maximum and minimum reward values, as the lower the distance, the less variability. Conversely, a higher distance means more variability in the learning curve. Hence, it can be illustrated how good the agent's stability level was. The Figure's plot lines were convoluted using a combination of filtering processes with a window of 10 and 30 for the Cliff Walking and Cart Pole environments, respectively.



(a) Agent's collected reward values.

(b) Number of times the agent reached an unsafe state.

Fig. 5. Average results along with the maximum and minimum values over 250 episodes of executing the Cliff Walking scenario for 10 agents. CA corresponds to contextual affordance.

The convolution of the data allows smoother graphs data, facilitating results analysis and interpretation.

A. Cliff Walking environment

For the Cliff walking scenario, an artificial neural network was implemented to infer the action effect under CA. The model's architecture comprises one hidden layer with 190 units using a sigmoid activation function, 4 units in the input layer, and 1 unit in the output layer. The input layer accepts the state and action to infer and predict the unsafe state. The output layer also uses a sigmoid activation function. The entire model was trained under a 0.01 learning rate value.

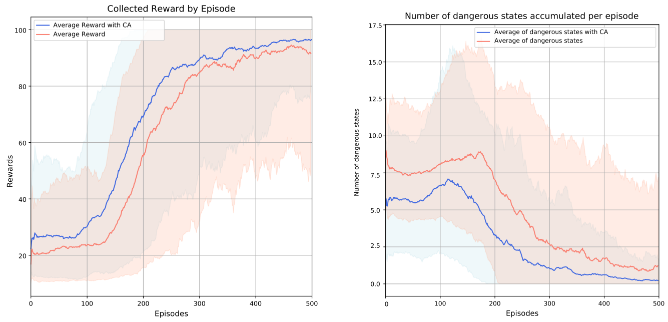
Observing the collected reward in Fig. 5(a), an improvement of 70.53% in the average collected reward when using CA can be noticed. The average reward value is reduced from -1432 to -422 at the very beginning of the training. This improvement reduces the time in achieving high reward value by avoiding unsafe states and being constant as the episodes progress. At episode 250, a 37.08% improvement is observed when using CA.

The number of unsafe state visits backs up the behaviour of using CA in reinforcement learning agents. When analyzing Fig. 5(b), it can be observed that the agent enters into dangerous states an average of 13.9 times on its vanilla implementation. When using CA, the number of visited unsafe states is reduced by 76.97%.

B. Cart Pole environment

As in the Cliff walking scenario, a neural network was trained to use CA with reinforcement learning. In this case, the model's architecture is composed of a hidden layer with 890 neurons, keeping the input and output layers the same size as in the previous environment. The hidden and output layers perform a sigmoid activation function. In this case, the input layer accepts the current pole's angle, the angular velocity, and the action to predict an unsafe state. The learning rate was set to 0.01.

The results shown in Fig. 6(a) illustrate the CA comparison against the vanilla implementation trained over ten times. Contextual affordance performs 35.41% better, obtaining an



(a) Agent's collected reward values.

(b) Number of times the agent reached an unsafe state.

Fig. 6. Average results along with the maximum and minimum collected values over 500 episodes of executing the Cart Pole scenario for 10 agents. CA corresponds to contextual affordance.

average reward of 26, versus the vanilla implementation's 19.2. With the implementation of contextual affordance, the increase in the collected reward is not particularly remarkable compared to the Cliff Walking scenario. However, this improvement is maintained constant throughout the 500 episodes. Respect the number of visits to unsafe states in Fig. 6(b). CA use represents a decrease of 43.95% in the average number of times the agent falls into a dangerous state, reducing from 9.1 to 5.1.

The observed high variance in training curves for both models, given the simplicity of the testing environments, suggests that fine-tuning the hyper-parameters might be beneficial. While no specific hyper-parameter validation was conducted, a systematic approach was taken to explore various popular hyper-parameter settings until finding configurations that suited both scenarios. The parameters shown in Table II, were chosen empirically based on their ability to facilitate learning in both controlled and full scenarios.

V. Conclusions

Contextual affordance has proven to be a valuable tool for safe explorative robotic agents. The results in the Cliff Walking and Cart Pole scenarios showed that contextual affordance provides significant improvements in terms of collected reward and prevention of unsafe states at early episodes.

The integration of contextual affordance into the reinforcement learning framework supports the action exploration phase, preventing the agent from falling into unsafe states. Consequently, contextual affordance enables safe exploration of actions while aiming to reach the optimal policy. The results indicate in general higher cumulative rewards by avoiding entry into penalized states, which would otherwise diminish the final reward. Although in the Cart Pole scenario the improvement in the cumulative reward when using contextual affordance is less, still a considerable improvement is observed in the context of safe exploration, especially in the initial episodes of the agent.

In general, contextual affordance offers a complementary approach to RL, providing external support to the agent to make informed decisions and avoid harmful situations. Its successful application depends on each scenario's context and characteristics, offering new opportunities to address complex real-world problems and promising to drive further advances in artificial intelligence and robotics.

For future work, it is proposed to extend the evaluation of the contextual affordance model using deep reinforcement learning (Deep RL). This extension aims to address the evaluation of more complex scenarios involving continuous states. Additionally, the objective is to derive metrics, including convergence speed, cumulative rewards, and the prevention of unsafe states, for analyzing the performance of the contextual affordance model in scenarios with continuous states. This comprehensive set of metrics is intended to provide a thorough understanding of the model's effectiveness in scenarios with increased variables or complexity within the context of safe exploration.

Moreover, given the challenge of generalizing to unseen cliff position in the cliff walking environment, it is recommended to utilize the presence of a cliff in the neighboring cells as a state descriptor, rather than relying solely on coordinates. This approach would facilitate the agent's ability to generalize to unseen cliff positions, thereby enhancing its adaptability and overall performance.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] P. Geibel, "Reinforcement learning with bounded risk," in *In Proceedings of the Eighteenth International Conference on Machine Learning*, Citeseer, 2001.

[3] M. Pecka and T. Svoboda, "Safe exploration techniques for reinforcement learning–an overview," in *International Workshop on Modelling and Simulation for Autonomous Systems*, pp. 357–375, Springer, 2014.

[4] F. Cruz, G. I. Parisi, and S. Wermter, "Learning contextual affordances with an associative neural architecture.," in *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pp. 665–670, 2016.

[5] W. J. R. Henderson, A. Valero-Kerrick, A. Garcia-Nevarez, and K. A. G. Biddle, *Early childhood education: Becoming a professional*. Sage Publications, 2013.

[6] J. J. Gibson, "The ecological approach to the visual perception of pictures," *Leonardo*, vol. 11, no. 3, pp. 227–235, 1978.

[7] T. E. Horton, A. Chakraborty, and R. S. Amant, "Affordances for robots: a brief survey," *AVANT. Pismo Awangardy Filozoficzno-Naukowej*, vol. 2, pp. 70–84, 2012.

[8] F. Cruz, S. Magg, C. Weber, and S. Wermter, "Improving reinforcement learning with interactive feedback and affordances," in *4th International Conference on Development and Learning and on Epigenetic Robotics*, pp. 165–170, IEEE, 2014.

[9] F. Cruz, S. Magg, C. Weber, and S. Wermter, "Training agents with interactive reinforcement learning and contextual affordances," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 271–284, 2016.

[10] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[11] A. Ayala, C. Henríquez, and F. Cruz, "Reinforcement learning using continuous states and interactive feedback," in *Proceedings of the 2nd International Conference on Applications of Intelligent Systems*, pp. 1–5, 2019.

[12] D. Michie and R. A. Chambers, "Boxes: An experiment in adaptive control," *Machine intelligence*, vol. 2, no. 2, pp. 137–152, 1968.