



Proxemic behavior in navigation tasks using reinforcement learning

Cristian Millán-Arias¹ · Bruno Fernandes² · Francisco Cruz^{3,4}

Received: 14March2022 / Accepted: 6July2022 / Published online: 16August2022
© The Author(s)2022

Abstract Human interaction starts with a person approaching another one, respecting their personal space to prevent uncomfortable feelings. Spatial behavior, called proxemics, allows defining an acceptable distance so that the interaction process begins appropriately. In recent decades, human-agent interaction has been an area of interest for researchers, where it is proposed that artificial agents naturally interact with people. Thus, new alternatives are needed to allow optimal communication, avoiding humans feeling uncomfortable. Several works consider proxemic behavior with cognitive agents, where human-robot interaction techniques and machine learning are implemented. However, it is assumed that the personal space is fixed and known in advance, and the agent is only expected to make an optimal trajectory toward the person. In this work, we focus on studying the behavior of a reinforcement learning agent in a proxemic-based environment. Experiments were carried out implementing a grid-world problem and a continuous simulated robotic approaching environment. These environments assume that there is an issuer agent that provides non-conformity information. Our results suggest that the agent can identify regions where the issuer feels uncomfortable and find the best path to approach the issuer. The results obtained highlight the usefulness of reinforcement learning in order to identify proxemic regions.

Keywords Cognitiveagents · Proxemics · Reinforcementlearning

1 Introduction

During a human interaction, people should feel comfortable when they perceive that other are approaching. Therefore, the approaching person should respect the

intimate area. Proxemics is the study of spatial behavior, concerned with territoriality, interpersonal distance, spatial arrangements, crowding, and other aspects of the physical environment that affect behavior. The term was coined by Hall et al. [1]. He proposed a fixed measure of personal space, a set of regions around a person to delimit the acceptable distance to interact with other people.

In recent years, human-agent interaction has gathered popularity in the scientific community [2]. Furthermore, the humanization of agents is an expected event, given technological advances and human nature. Thus, optimal interaction is necessary for both the agent and the person [3].

Reinforcement Learning (RL) is a learning paradigm that tries to solve the problem of an agent interacting with the environment to learn a desired task autonomously [4]. The agent must sense a state from the environment and take actions that affect it to reach a new state. The agent receives a reward signal from the environment that they try to maximize throughout the learning for each action taken. The agent takes actions from their own experience, or can be guided by an external trainer that provides feedback [5, 6].

✉ Cristian Millán-Arias
ccma@ecomp.poli.br

Bruno Fernandes
bjtf@ecomp.poli.br

Francisco Cruz
f.cruz@unsw.edu.au

¹ Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brazil

² Present Address: Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brazil

³ Present Address: School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

⁴ Escuela de Ingeniería, Universidad Central de Chile, Santiago, Chile

Proxemic behavior has been used in different areas with cognitive agents. For example, human-robot interaction studies how people behave in the presence of an artificial agent or robot and how the agent perceives the personal space [7, 8]. Moreover, machine learning has been used to study how artificial agents sense the personal space of other cognitive agents, and it has been used to identify and learn personal space [9, 10]. These implementations consider that the personal space is fixed and that the agent previously knows such space. However, these assumptions are not necessarily accurate in real-world scenarios; personal space is different for each person and unknown. Different characteristics dynamically modify the personal space, such as culture, family environment, the territory where they live, and previous experiences, making proxemics' identification more complex and requiring external information to estimate it. In navigation tasks, RL agents learn a policy to control their movement in space and move. In the sense of approximation, they perceive the cognitive agent as one more obstacle or a goal, so the distance relationship is limited to that allowed by the agent to avoid colliding. In this work, we study how a RL agent learns to approach another cognitive agent in two proxemics-based environments. To address this problem, we implement Q-Learning and advantage actor-critic (A2C) algorithms in a modified grid-world and a simulated robot approaching problem, respectively, where a cognitive agent, the issuer, gives information to the learning agent. Our study is limited to experiments with fixed initial values, i.e., the RL agent remains in the same position at the beginning of each episode. Also, cognitive agents and their personal space remain fixed during training. However, we find that the trajectories of an RL agent may be able to identify the proxemic region of cognitive agents. Thus, the agent learns a policy to move in an environment and finds a limit to which to move.

This paper is organized as follows: In Sect. 2, we present related works. Section 3 introduces the basics of RL, Q-learning and advantage actor-critic. In Sect. 4, we describe the environments with proxemic behavior in cognitive agents, a modified grid-world and the robot approaching based on ship steering. Experiments, results and discussion are described in Sect. 5. Finally, Sect. 6 presents the main conclusions and describes future research.

2 Related works

The interaction process is an animal and human characteristic that allows the creation of social bonds for survival or with a specific objective. This process generally has an approach, action execution, and mutual response. However, the mere approach is already an interaction in itself. The

execution is the act of approaching, and the response is how the other reacts to the action of approaching. In nature, the animals have a behavior based on the species and the crowd, when each group of animals remain in a territory because of the feed source or security [11, 12]. In the interaction process, each human uses their personal space and perceives the space of another being. The way how each human uses and perceive depends on the experience of previous interactions or contact [1]. It is clear that the experience is unique in the sense that each human creates relations with other beings, based on nature, beliefs, culture and society. These aspects constitute a set of rules that allow an effective interaction avoiding disagreement between the parties.

Including artificial agents in a human environment humanizes these entities, in the sense that they have to behave according to the rules that people build from their experience. Figure 1 shows how the human-robot interaction process is in the presence of proxemic behavior. The interaction process includes the robot perception, where the artificial agent perceives the reactions of the person, the robot learning and robot performance, where the robot uses human information to learn the task and perform it and, finally, the human perception and reaction, where the human reacts to the performed action by the agent.

2.1 How the human perceives an artificial proxemic behavior?

It is essential to know how a person perceives the artificial agent. Previous works have found that people tend to maintain social aspects in environments that involve robots or virtual agents [13]. For instance, Li et al. [14] has shown

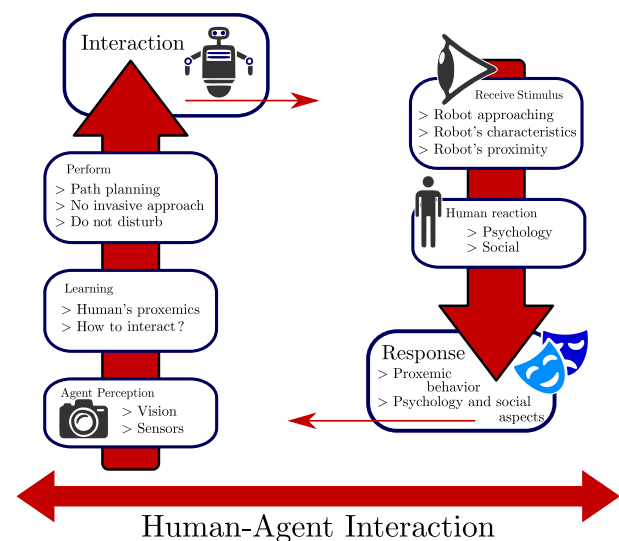


Fig. 1 General process of interaction with proxemic behavior: Human perception, Human reaction, Human response, Robot perception, Learning, and approach (task)

that humans maintained social distance in virtual environments. In addition, people responded to the actions of virtual agents and tried to remain at a comfortable distance. Humans not only responded to the movement of other cognitive agents, but they also needed another signal to react according to their personal space. Another type of interaction could change the personal space; for instance, when the virtual agent made a wave, the human reacted and modified their social space [15].

In social environments, humans maintain appropriate distances between themselves. Furthermore, this behavior is extended to virtual environments, where a human maintains a relative distance according to the perception in this environment [16]. Kastanis and Slater [17] investigated how a learning agent could drive human participants to a certain previously established position in a virtual space, exploiting human proxemic behavior. In their work, a reinforcement learning approach was used for a robot to learn how to influence people to move to another position. The authors concluded that the goal was affected by human behavior. The agent learned to approach the participant when they were too far and exploited the proxemics to carry the person back to the goal position. Another case is to use an action that maintains the attention of the human/agent. For instance, Kastanis and Slater [17] used wave action to get the participants' attention when it was too far from them. Millan et al. [18] used PING action to communicate with a learning agent and an informative issuer.

2.2 How do artificial agents identify non-verbal aspects?

In non-verbal communication, there are different ways of identifying social cues to predict discomfort, agreement, attitude, or another behavioral trait of interest. Individually, each non-verbal aspect gives some information about the behavior of the human being. Ponce-Lopez et al. [19] applied a non-invasive ambient intelligence framework for analysis of non-verbal communication in conversation settings. The learning was a binarized classification problem, where the authors used Adaboost, Support Vector Machines (SVM), and two kinds of Artificial Neural Networks (ANN), in particular Cascade-Forward (CF) and FeedForward neural networks (FF) [20, 21]. Another approach to identifying nonverbal communication was presented in [22]. The authors propose a computational framework for nonverbal communication for human-robot interaction, where a storyteller had to recognize attention from a listener. The main objective was to recognize that attention encodes social-emotional states through nonverbal behaviors.

Concerning proxemics, several works try to predict or estimate the social space from humans. Different strategies

in the human-robot interaction area suggest extracting information to construct this space out of the robot control. For instance, personal space can be related to an uncomfortable level that indicates how much the agent discomforts the human. Generally, this discomfort level is estimated based on information about the demography, sociology, or psychology aspects from participants in a study. For instance, Kosinski et al. [23] performed a study with participants to obtain information about demographic aspects. The study was performed in a robotic environment, where the participants stopped the robot when they felt discomfort. The robot's position from different angles was captured and used to construct the comfort level.

On the other hand, machine learning is used to find relationships between a person's level of discomfort during the interaction and personal information. Gao et al. [24] used different deep learning neural networks architectures to predict comfortable human-robot proxemics based on the personal information obtained from each participant. The authors drove the problem as a regression problem, where the response was the numerical discomfort and the regressors variables were the values of distance, angles, and demographic aspects. Kosinski et al. [23] used a fuzzy dataset-based model, to encode the study information parametrizing rules to reflect user preferences in the sense of personal space. Other approaches intend to exploit the image information to estimate distance. This methodology can carry several issues due to the camera, quality, and configuration of images. Seker et al. [25] proposed a benchmark to evaluate the social distance. For this, the authors created a dataset with measured pair-wise social distance under different camera positions. The authors used YOLOv4 [26] and OpenPose [27] to detect and identify skeletal points from the people.

2.3 How do artificial agents learn proxemic behavior?

In environments that involve humans' proxemic behavior, an agent has to find the best path to reach a specific place navigating among humans. The learning agent must be socially aware that its movement cannot perturb human activity [28, 29]. In general, the agent moves in the social or public space, avoiding getting closer to the people. However, several works show that the estimated paths invade the personal space from proxemic theory [30]. These findings suggest that motion-based in proxemic behavior has poorer performance than methods that imitate human behavior. Nevertheless, the works do not consider that humans' proxemic behaviors can change based on external factors or the situation, as in the navigation process. Charalampous et al. [31] proposed a robot framework to navigate in a human-populated environment. This

approach considered that the robot exhibits socially accepted behavior by considering the actions being performed by individuals. The methodology was applied in indoor areas and working environments, where people gathered in groups of two or three, but not in a congested environment. The robot navigated based on a known pre-computed 3D metric map (constructed offline) during the learning. The authors used a deep learning action recognition with HTM, consequently, an SVM to classify each action. Luber et al. [30] suggested a learning model from observations of humans, i. e., the learning agent learned through human relative motion behavior. The authors used an unsupervised way to classify the relative motion prototypes in the methodology. Also, they used a distance function with an asymmetric Dynamic Time Warping algorithm (DTW) to ensure that the motion sequences did not differ in duration and relative speed. They showed that their methodology can imitate human motion behavior, and its performance was better than learning using the proxemic theory (Hall's theory).

During navigation, it is expected that cognitive agents tend to intrude the personal space, differently to the interaction where this nearness is acceptable only for people of great social acceptability as good friends or family members. Luber et al. [30] used relative motion prototypes (RMP) and showed that social acceptability or comfort in humans is higher when the learning agent exhibits the same behavior as a human. They used the person orientation as a criterion to quantify and distinguish what they defined here to be social context. Besides, they compared Hall's theory of proxemics and highlighted that invading personal space appears frequent during navigation. However, the personal space was considered fixed, and it reduced its size during the trajectory due to the social situation. Feldmaier et al. [32] combined an emotion model with the SLAM algorithm to allow the robot to give non-verbal feedback to a user about the internal processes. The learning agent explored the environment using the SLAM algorithm while simultaneously appraising its current progress with emotions. The estimations of the SLAM were used as input in an appraisal model to perform the Stimulus Evaluation Checks (SECs) that were independent modules in which the agent did its subjective assessment of the situation based on personal needs, goals, and values. Then, a categorization module was used to map the pleasure and arousal space defined by Russell [33].

Other works consider the robot as part of a human group, in the sense that the robot behaves similarly to the individuals or behaves according to social rules. Fuse et al. [34] proposes a robotic model that enables identifying the robot's position in a social people group. Furthermore, the robot considered the change of personal space during the navigation process. Also, they examined if a robot could

imitate the trajectory of humans. They used an RL framework in their implementation, considering a value function to the states and actions and a value function to the distance. The latter function was used to learn the agent's physical distances and determine its physical position. They concluded that the learning agent maintained a behavior like humans while it was approaching them. In addition, a group of participants assessed the agent trajectories and determined that they were similar to human trajectories. The main goal of their work was that the learning agent does not use trajectory demonstrations from humans to imitate their behavior. Our work studies the behavior of an agent in an environment with proxemic behavior, where it learns trajectories that do not invade the personal space of an issuer. Table 1 shows a comparison of some works mentioned above and the objective of each one.

In Silva and Romero [35], the authors proposed a framework of sharing attention, where a learning agent was provided the capacity of joint attention with a human. Their framework used a combination of relational reinforcement learning (RRL) and recurrent neural network (RNN) with state classification, the so-called RLSSACTG. The main idea was to maintain the attention in a human based on gaze behavior, changing the attention to an object when this appeared, and backing to pay attention to the human. They studied the framework's performance in a simulated environment, where a learning agent faced a human agent. They compared the RL (Q-learning) approach, ETG algorithm [36], and RLSSACTG. The obtained results showed that architecture is a potential tool to control sociable robots.

Inverse reinforcement learning (InvRL) is commonly used in navigation problems. The main idea is learning from demonstration, where the agent tries to find the patterns of experts based on their demonstration behavior. This approach facilitates the agent to learn how to navigate among people because it recognizes human behavior in demonstrating paths [28]. InvRL is generally used to learn the cost function associated with the path planning algorithm or the navigation controller. InvRL shows better results than the policy based on proxemic theory because cost functions can be used in different human environments [29].

3 Reinforcement learning framework

Reinforcement learning is a learning approach that involves an autonomous agent learning from interactions with its environment to achieve a goal [4]. The agent must be able to learn from its own experience by selecting actions that affect the environment, reaching situations that allow it to complete the task. In this approach, the learning

Table 1 Comparison of different works and its methods to estimate trajectories. Our study shows that it is possible to identify the issuer’s proxemic region based on the agent’s trajectories

Author	Method	Objective
Luber et al. [30]	RMP + Social context	To reproduce human relative motion
Feldmaier et al. [32]	SLAM + Emotional model	Generate emotional map by navigate
Fuse et al. [34]	RL	To learn physical distance
Our study	RL	To learn non-invasive trajectories

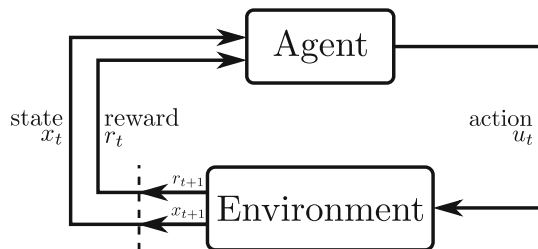


Fig. 2 Interaction between the agent and the environment in the Reinforcement Learning context. Figure adapted from Sutton and Barto’s book [4]

agent receives a numerical reward signal from the environment it that attempts to maximize [37].

The agent and the environment interact at each time step $t = 0, 1, 2, \dots$. At each step, the learning agent receives a representation of the state of the environment $x_t \in X$, and of the action selected by the agent $u_t \in U(x_t)$, where X is a set of the all possibles states, and $U(x_t)$ is a set of the actions available in the state x_t . Eventually, as a result of performing an action, the agent receives a scalar reward $r_{t+1} \in \mathbb{R}$ and reaches a new state x_{t+1} [4]. Figure 2 shows the diagram of the interaction between the agent and the environment, in the context of RL.

At each time step, the agent relates states to probabilities to select each possible action. These relations are called the agent policy and denoted by π_t , where $\pi_t(u_t | x_t)$ is the probability to select u_t given the current state x_t at time t . The RL method specifies how the agent changes the policy as a consequence of its experience. Thereby, the goal of the reinforcement learning agent is to approximate a function $\pi : X \times U \rightarrow (0, 1)$ that maximizes the total amount of reward it receives over the long run.

A value function represents an estimation of how good a particular action from that state is in terms of future expected reward [38]. The *state-value function* of x , under all the actions, is the expected return of the discounted sum starting in the state x and following a policy π . Formally, we can define it by [4, 38]

$$V^\pi(x) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid x_t = x \right], \tag{1}$$

where $E_\pi[\cdot]$ denotes the expected value given that the agent follows the policy¹ π .

Similarly, it is defined as an *action-value function* as the value of taking action u in the state x under the policy π . Roughly speaking, it is the expected return starting from state x , taking action u and following the policy π [4]

$$Q^\pi(x, u) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid x_t = x, u_t = u \right]. \tag{2}$$

In addition, the value functions satisfy a recursive property. The expression in (1) can be defined recursively in terms of the so-called Bellman equation [39]. It is denoted as the expected return in terms of the immediate reward and the value of the next state, defined formally by

$$V^\pi(x) = \sum_U \pi(u | x) \sum_X P(x' | x, u) \left[\rho(x, u, x') + \gamma V^\pi(x') \right] dx' du, \tag{3}$$

where x is the current state, u is the selected action and x' is the new state reached by performing the action u in the state x . The function $Q(x, u)$ in (2) can also be defined recursively in terms of the Bellman equation.

3.1 Q-learning

The Q-learning algorithm [40] is an off-policy method based on TD learning. The one-step Q-learning is a simple algorithm, in which the update of the value $Q(x_t, u_t)$ is carried out using the value $\max_{u \in U(x_{t+1})} Q(x_{t+1}, u_t)$. The core of the algorithm is based on Bellman equation as a simple value iteration update, using the weighted average of the old value and the new information of state:

$$Q(x_t, u_t) \leftarrow Q(x_t, u_t) + \alpha \left[\rho_{t+1} + \gamma \max_{u \in U(x_{t+1})} Q(x_{t+1}, u_t) - Q(x_t, u_t) \right], \tag{4}$$

where α is the learning rate, and γ is the discount factor. The value of $Q(x_t, u_t)$ estimates the action-value after applying action u_t in state x_t . The learned action-value function, Q , directly approximates the optimal action-value function Q^* , independent of the policy being followed [4].

¹ A summation or integrals define the expected value according to the nature of the sets X and U .

3.2 Advantage actor-critic

The advantage actor-critic (A2C) algorithm is an on-policy method based on TD actor-critic that keeps a separate memory structure to represent the policy, independent of the value function [41]. The agent is separated into two entities: the actor and the critic. The policy takes the role of the actor, selecting actions in each iteration. The critic, commonly a state-value function, evaluates or criticizes the actions performed by the actor [42]. Figure 3 shows the schematic structure of the general actor-critic algorithm.

In each iteration, the critic values each action through TD error:

$$\delta_t = r_{t+1} + \gamma V(x_{t+1}) - V(x_t). \tag{5}$$

Policy-gradient methods [43] are a methodology to approximate functions in RL. These methods are the most popular class of continuous action RL algorithms [44]. With this approach, a stochastic policy is approached through an approximation function, independent of the value function, with its parameters. A measure is used to improve the performance of the policy and adjusts the parameters.

In this context, the policy-gradient framework uses a stochastic policy π , parametrized by a column vector of weights $v \in \mathbb{R}^{N_a}$, for $N_a \in \mathbb{Z}$. $\pi(u | x)$ denotes the probability density for taking action u in the state x . The objective function $\Gamma(\pi)$ maps policies to scalar measure of performance, defined by

$$\Gamma(\pi) = \int_X d^\pi(x) \int_U \pi_v(u | x) Q^\pi(x, u) dudx, \tag{6}$$

where $d^\pi(x) := \int_X \sum_{t=0}^\infty \gamma^{t-1} P(x | x_0, u)$ is the stationary distribution of the discounted states occupancy under π and

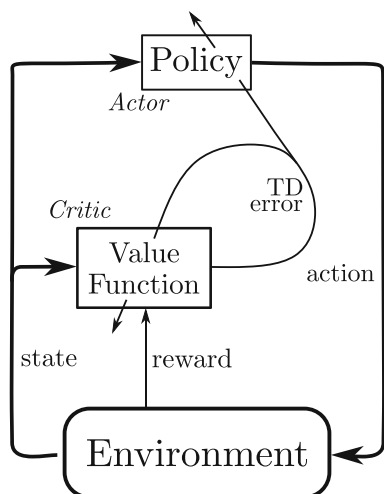


Fig. 3 Schematic overview of an actor-critic algorithm. Taken from Sutton and Barto’s book [4]

$Q^\pi(x, u)$ is the action value function. The basic idea of policy-gradient methods is to adjust the parameter v of the policy in the direction of the gradient $\nabla_v \Gamma(\pi)$:

$$v_{t+1} - v_t \approx \alpha_v \nabla_v \Gamma(\pi). \tag{7}$$

The fundamental result of these methods is the policy-gradient theorem [43], which defines the gradient as

$$\nabla_v \Gamma(\pi) = \int_X d^\pi(x) \int_U \nabla_v \pi_v(u | x) Q^\pi(x, u) dudx. \tag{8}$$

Let $h_\theta : X \times U \rightarrow \mathbb{R}$ be an approximation of the value function $Q^\pi(x, u)$ with the parameter $\theta \in \mathbb{R}^{N_c}$, for $N_c \in \mathbb{Z}$, so that it does not affect the unbiasedness of the policy-gradient estimate. To find a close approximation of $Q^\pi(x, u)$ by $h_\theta(x, u)$, the parameter θ must be found to minimize the quadratic error of the approximation as follows:

$$\epsilon_t^\pi(x, u) = \frac{1}{2} [Q^\pi(x, u) - h_\theta(x, u)]^2. \tag{9}$$

The gradient of the quadratic error can be used to find an optimal value of θ . Considering the approximation of Q^π by h_θ , Eq. (8) is expressed as:

$$\nabla_v \Gamma(\pi) = \int_X d^\pi(x) \int_U \nabla_v \pi_v(u | x) h_\theta(x, u) dudx. \tag{10}$$

This approximation is acceptable if $\nabla_\theta h_\theta(x, u) = \nabla_\theta \log(\pi_v(u | x))$ is satisfied.

Under this approach, the expected value of h_θ given policy π is zero, indicating the value function has zero mean in each state. The most convenient is to approximate an advantage function $A^\pi(x, u) = Q^\pi(x, u) - V^\pi(x)$ instead of $Q^\pi(x, u)$ [45]. This implies that the approximation function only represents the relative value of an action u in some state x and not the absolute value of Q^π [46]. The value function $V^\pi(x)$ is a baseline in the advantage function, such that the variance of the policy-gradient is minimized [47]. In other view, as

$$Q^\pi(x, u) = E_\pi[r_{t+1} + V^\pi(x_{t+1}) | x_t = x, u_t = u], \tag{11}$$

then, according to Sutton and Barto [4]:

$$A^\pi(x, u) = E_\pi[r_{t+1} + (V^\pi(x_{t+1}) - V^\pi(x_t)) | x_t = x, u_t = u], \tag{12}$$

that is the expected value of the TD error (5). Thus, h_θ would approximate a value function $V^\pi(x)$, and the gradient in (10) takes the form of

$$\begin{aligned} \nabla_v \Gamma(\pi) &= \int_X d^\pi(x) \int_U \nabla_v \pi_v(u | x) \hat{\delta}_t dudx \\ &= E_\pi[\hat{\delta}_t \nabla_v \log(\pi_v(u | x))], \end{aligned} \tag{13}$$

where $\hat{\delta}_t = r_{t+1} + \gamma h_\theta(x_{t+1}) - h_\theta(x_t)$, and $h_\theta(x) : X \rightarrow \mathbb{R}$ the function approximation of value function $V^\pi(x)$.

In the context of A2C, let $h_\theta(x) = V_\theta(x)$ be the approximate state value function from the critic. The update from the value function is through θ . The parameter θ is adjusted by the gradient of the quadratic error (9) as follows:

$$\theta_{t+1} = \theta_t + \alpha_\theta \hat{\delta}_t \nabla_\theta V_{\theta_t}(x_t),$$

where $\alpha_\theta > 0$ is a step-size parameter of the critic. It is clear that the gradient of the quadratic error is the gradient of the approximation scaled by the TD error. On the other hand, the policy, that represents the actor, is updated based on the gradient in (12) as follow:

$$v_{t+1} = v_t + \alpha_v \hat{\delta}_t \nabla_v \log(\pi_v(u | x)),$$

where $\alpha_v > 0$ is a step-size parameter of the actor.

$$\rho_{grid} = \begin{cases} -1.0 & \text{if conditions C1 or C2} \\ -0.8 & \text{if the agent reaches the uncomfortable region} \\ -0.4 & \text{if the agent sends PING out of the target region} \\ +1.0 & \text{if condition C3} \\ -0.1 & \text{if the agent reaches another state} \end{cases} \tag{15}$$

new action, PING, is added to the four traditional ones (UP, DOWN, LEFT, RIGHT). This action represents a communication signal with the issuer, i.e., the approaching agent sends a ping to ask the issuer if it is in the target region.

The task finishes in three conditions:

- *Condition C1*: When the agent reaches the issuer.
- *Condition C2*: When the agent sends ping five times out of the target region.
- *Condition C3*: When the agent sends ping in the target region.

The reward function is defined as

$$\rho = \rho_{issuer} + \rho_{grid}, \tag{14}$$

where ρ_{issuer} is a numerical reward given by the issuer when the agent performs the PING action, and ρ_{grid} is the environment reward defined as

4 Proxemics behavior in cognitive agents

In this section, we present different scenarios where agents confront another artificial agent with a proxemic characteristic. These scenarios are modifications of state-of-the-art environments, mimicking a proxemics human scenario, where cognitive agents give signals or react to the agent behavior.

4.1 Proxemics grid-world problem

We propose a modified version of the discrete grid-world problem [18]. In this environment, an issuer is placed on one fixed state and is responsible for giving a signal of disagreement when the learning agent is too close. Two regions are defined around the issuer, the uncomfortable region and the target region. The uncomfortable region comprises the states that make up a square around the issuer. In this environment, the uncomfortable region is only an area of negative reward. Similarly, the target region comprises the states that make up a square around the uncomfortable region. Figure 4 shows the proxemics grid-world domain with uncomfortable and target regions. A

In our experiments, we use a 10×12 grid-world, the issuer is placed in (6, 8) and is fixed during the training. Each agent starts the training in the superior left corner of the grid (0, 0).

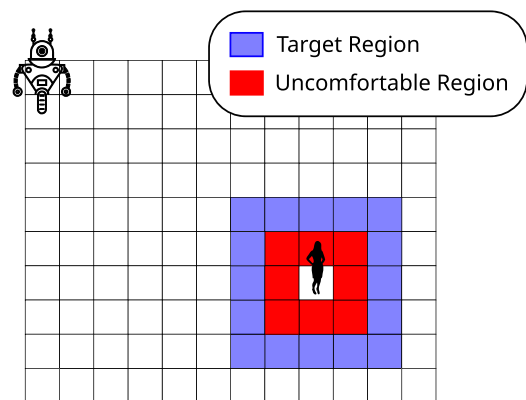


Fig. 4 Proxemics grid-world domain. A robot starts in an initial position. The issuer is placed in a fixed position and has two regions, the uncomfortable region (red) and the target region (blue) (color figure online)

4.2 Robot approaching problem

We present a robot approaching problem based on the ship steering problem. This is an episodic problem [48, 49] where a ship has to maneuver at a constant speed through a gate placed in a fixed position. At each time t , the state of the ship is given by its coordinate position x and y , orientation θ , and actual turning rate $\dot{\theta}$. The action is the desired turning rate of the ship r . In each episode, the goal is to learn a sequence of actions to get the ship through the gate in the minimum amount of time.

Based on the mechanic of the ship steering, we propose the robot approaching problem to include proxemic behavior (see Fig. 5). In our environment, a robot replaces the ship, that is the learning agent. The issuer replaces the gate, giving the signal or information when the agent is too close. Concerning the proxemic problem, the issuer, placed in a fixed position, has an individual space that avoids the robot hitting it.

To explore the behavior of the learning agent when the issuer gives any information during the learning, we propose two modifications on our environment. These modifications involve the personal space area and the direct information given by the issuer.

4.2.1 Robot approaching in uncomfortable region

In this environment, two regions are defined around the issuer, the uncomfortable and the target regions (similar to the grid-world problem). The uncomfortable region is a circle area of radius r_u around the issuer that mimics the intimate space of a person. The agent receives a negative reward if it invades this region. The target region (or social region) is a circle area of radius r_T ($r_T \gg r_u$) that contains the uncomfortable region (two concentric regions). In our implementation, the values of r_u and r_T are 10 and 20 meters, respectively. Figure 5 shows the robot approaching domain with the uncomfortable and target regions.

Similarly to the grid-world problem, we add the probability of sending PING, due to the PING being a binary action. Then, in each time step, the agent selects a two dimensional action, the turning rate and the probability of sending PING in this position.

In our experiments, we use the logit of the probability as action:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right). \tag{16}$$

The logit modifies the domain of the action, allowing that the action concerning the PING has the same domain that the turning rate. The task finishes under the next conditions:

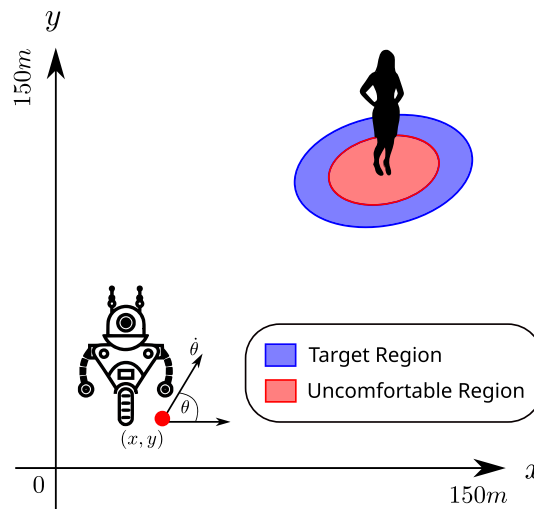


Fig. 5 The robot approaching domain. A robot starts in an initial position, direction and turning rate. The issuer is placed in a fixed position and has two regions, the uncomfortable region (red) and the target region (blue) (color figure online)

- *Condition C1*: When the agent is out of bounds.
- *Condition C2*: When the agent reaches the uncomfortable region.
- *Condition C3*: When the agent sends PING in the target region.

The reward function is defined as

$$\rho = \begin{cases} -100 & \text{if conditions C1 or C2} \\ -4K & \text{if the agent sends PING out of target region} \\ -1 & \text{if another state} \\ 0 & \text{if condition C3} \end{cases} \tag{17}$$

where $K = d(A_{(x,y)}, I_{(x,y)})(x_I^2 + y_I^2)^{-\frac{1}{2}}$, and $d(A_{(x,y)}, I_{(x,y)})$ is the euclidean distance between the agent, in the position (x_A, y_A) , and the issuer, in the position (x_I, y_I) .

4.2.2 Robot approaching with disagreement area

In this environment, the field is divided in two areas, the disagreement area and the agreement area. The disagreement area is a circle area of radius r_d with the center in the issuer. This area is composed of two subregions, the disagreement no stop region and the disagreement stop region. In the former, the agent is free to move in everywhere, but receives a negative reward in each step. In the latter, the agents stop immediately when they reach there. The disagreement area mimics the private area from a person when a no intimate interaction occurs, i.e., when the two agents (learning agent and issuer) should be too close to occur the interaction. The disagreement stop region allows that the agent does not reach too close to the issuer. In our implementation, the value of r_d is 20 meters, and the

disagreement stop region has a radius of 10 meters. Finally, the agreement area includes the points outside of the disagreement area. In this area, the issuer is indifferent to the behavior of the learning agent, except when the agent stops. Figure 6 shows the robot approaching domain with disagreement areas, where blue is the agreement area, light red is the disagreement no stop region and dark red is the disagreement stop region.

With the aim at using the area as information to the learning agent, we consider the area as a part of the state. Then, at each time t , the state of robot is given by its coordinate position x and y , orientation θ , actual turning rate $\dot{\theta}$, and the disagreement region δ . Here, $\delta = 1$ if the robot is in the disagreement area, and $\delta = 0$ if the robot is in the agreement area. A new action, STOP, is added to the turning rate one. This action decides whether to stop or not the robot’s trajectory. Basically, the STOP action is the decision of the agent if it stops or continues moving in the environment. Similarly to the previous environment, we use the logit of the probability of STOP (see expression (15)) as action to maintain the same domain that the turning rate.

It is clear that the task ends when the learning agent selects STOP as action. However, there are other conditions to finish the task:

- *Condition C1*: When the agent is out of bounds.
- *Condition C2*: When the agent reaches the disagreement stop region.
- *Condition C3*: When the agent stops.

The reward function is defined as

$$\rho = \begin{cases} -100 & \text{if conditions C1 or C2} \\ -\rho_d & \text{if condition C3} \\ -50 & \text{if the agent is on disagreement no stop region} \\ -1 & \text{if another state} \end{cases}, \tag{18}$$

where $\rho_d = 100K(d(A_{(x,y)}, I_{(x,y)}) - r_d)$, $d(A_{(x,y)}, I_{(x,y)})$ is the euclidean distance between the agent (in the position (x_A, y_A)) and the issuer (in the position (x_I, y_I)), and

$$K = \begin{cases} (x_I^2 + y_I^2)^{-\frac{1}{2}} & \text{if } \delta = 0 \\ r_d & \text{if } \delta = 1 \end{cases}. \tag{19}$$

The $-\rho_d$ value punishes the agent when it stops far away from the issuer, and benefits the agent when it is too close to the disagreement area. The K factor weighs the reward based on which region the agent is in.

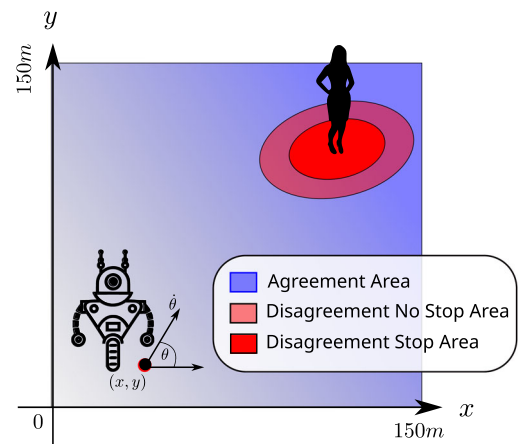


Fig. 6 The robot approach domain with disagreement area. A robot starts in the agreement area (blue) with an initial position, direction and turning rate. The issuer is placed in a fixed position and has two regions, the disagreement no stop region (light red) and the disagreement stop region (dark red) (color figure online)

5 Experimental results

This section presents the results obtained and the discussion of the research. To study the performance of the agents, we observe the behavior of the maximum action-value Q for the grid-world problem and the action-value Q for each action. Furthermore, for the continuous scenarios of the robot approaching problem, we study the behavior of the obtained discounted reward and the trajectories of the agents. We present a discussion about the behavior of the agents and what is the influence of each feature from the environment.

5.1 Grid-world problem

We perform the training of 100 agents with the Q-learning algorithm [40] in the modified grid-world problem. Each agent is trained during 10, 000 time-steps using the reward function presented in (13). We use ϵ -greedy as action selection policy. We set the values of both ϵ and the learning rate α at 0.6 and the discount factor γ at 0.9.

To study how the issuer influences the performance of the learning agent, we consider three scenarios where the issuer gives numerical information through the reward:

- Scenario S1: $\rho_{\text{issuer}} = 0$ in each PING action selection.
- Scenario S2: ρ_{issuer} is a random value in $(-1, 1)$.

- Scenario S3:

$$\rho_{\text{issuer}} = -\frac{d_1(P_{\text{agent}}, P_{\text{issuer}})}{d_1(P_{\text{start}}, P_{\text{issuer}})},$$

where $d_1(\cdot, \cdot)$ is the L_1 distance, P_{agent} , and P_{issuer} , and P_{start} are the position on the grid from the agent, the issuer and the start point. The factor ρ_{issuer} indicates that states further from the issuer have a low reward, and the denominator standardizes it on $(-1, 0)$.

Figure 7 shows the results of our experiments in the three scenarios. In terms of maximum Q-value per time step (Fig. 7a–c), there exists more variability when the issuer gives random rewards. However, according to the other scenarios, the maximum Q-value reaches values around 1 in several time steps. On the other hand, when the issuer gives rewards based on distance, the variability of Q-values is minor.

The Q-values for each action are more focused on the target region than on the other scenarios when the issuer gives rewards based on distance (Fig. 7d–f). This is expected due to the agent receiving additional information on how to move through the grid.

Concerning the PING, the issuer reward highlights the importance of giving PING in the target region, as shown in Fig. 7g–i. The graph shows how the states of the target region have only Q-values greater than zero. While with the lack of information (Scenario S1), states outside the target region have Q-values greater than zero.

5.2 Robot approaching problem

We study the performance of 20 learning agents in a continuous domain with proxemic behavior. We analyze the average collected discounted reward from a training and validation phase. The training phase consists of 1000 steps per episode, where weights are updated every 10 steps. The validation phase consists of 1000 steps after each episode, during these steps, the weights are not updated. In our experiments, the field size is 150×150 , the issuer is placed in (110, 110) and is fixed during the training. The agents are trained using advantage actor-critic (A2C) algorithm [41] for 10000 episodes, each one with 1000 time-steps. Each agent starts the training in the coordinate (1, 1). We use a Gaussian distribution to select random actions, with mean $\mu(x_t)$ depending on states. We used a multilayer perceptron (MLP) as function approximation with a hidden layer for the actor and the critic, the mean of distribution $\mu(x_t)$ and the value function $Q(x_t)$, respectively. The approximation is carried out as follows:

- For the value function $Q(x_t)$, an MLP with a hidden layer of 256 units, and an output layer with one unit is used.
- For the mean $\mu(x_t)$ of the policy, an MLP with a hidden layer of 256 units, and an output layer with two units (the turning rate and the logit) is used.

The input layer depends on the environment, *four* units are used for the robot approaching in uncomfortable region problem, and *five* units for the robot approaching with disagreement area problem. In both architectures, we apply as activation function the hyperbolic tangent (*tanh*) in the hidden layer, and in the output layer a linear activation. The learning rate is empirically set at $\alpha = 0.0007$, the discount factor γ is set at 0.9, and the standard deviation from policy at (1.5, 0.5), for turning rate and logit, respectively. The values of the weights in the neural network are randomly initialized from a uniform distribution based on Xavier weight initialization [50].

5.2.1 Uncomfortable region

First, we study the behavior of the learning agent in the robot approaching in uncomfortable region problem. Figure 8 shows the average discounted reward for 20 agents, where the red line is the average during the training and the blue line is the average during validation steps. We can observe that after 4000 episodes, the average reward is steady, however, after 6000 episodes, the average increases and finds a higher steady point. This behavior suggested that the weights of the value function $Q(x_t)$ reach a local minimum in these episodes.

To explore how the behavior of the trajectories is, we empirically select the agent with the highest reward in the validation phase. Figure 9 shows the trajectories of the agent during the task. We compute these trajectories with different start positions and orientations, and these only are shown when the agent reaches the target region and sends PING. Due to the fixed start position, the agent only learns to reach the target region in some part of the coordinate plane. However, the agent always finds the better way to finish the task independently of the start orientation. On the other hand, we observe that the agent always stops in the frontier of the target position. This behavior is due to the environment configuration, given that the agent stops immediately when it sends PING in the target region.

Figure 10 shows the trajectories with the probability of sending PING, where the blue region indicates a value of probability closer to zero, and the red region shows a value closer to one. We observe that the agent commences to send PING when it is close to target region. This behavior makes the space into two parts. In the first one, the agent does not communicate with the issuer, and in the second

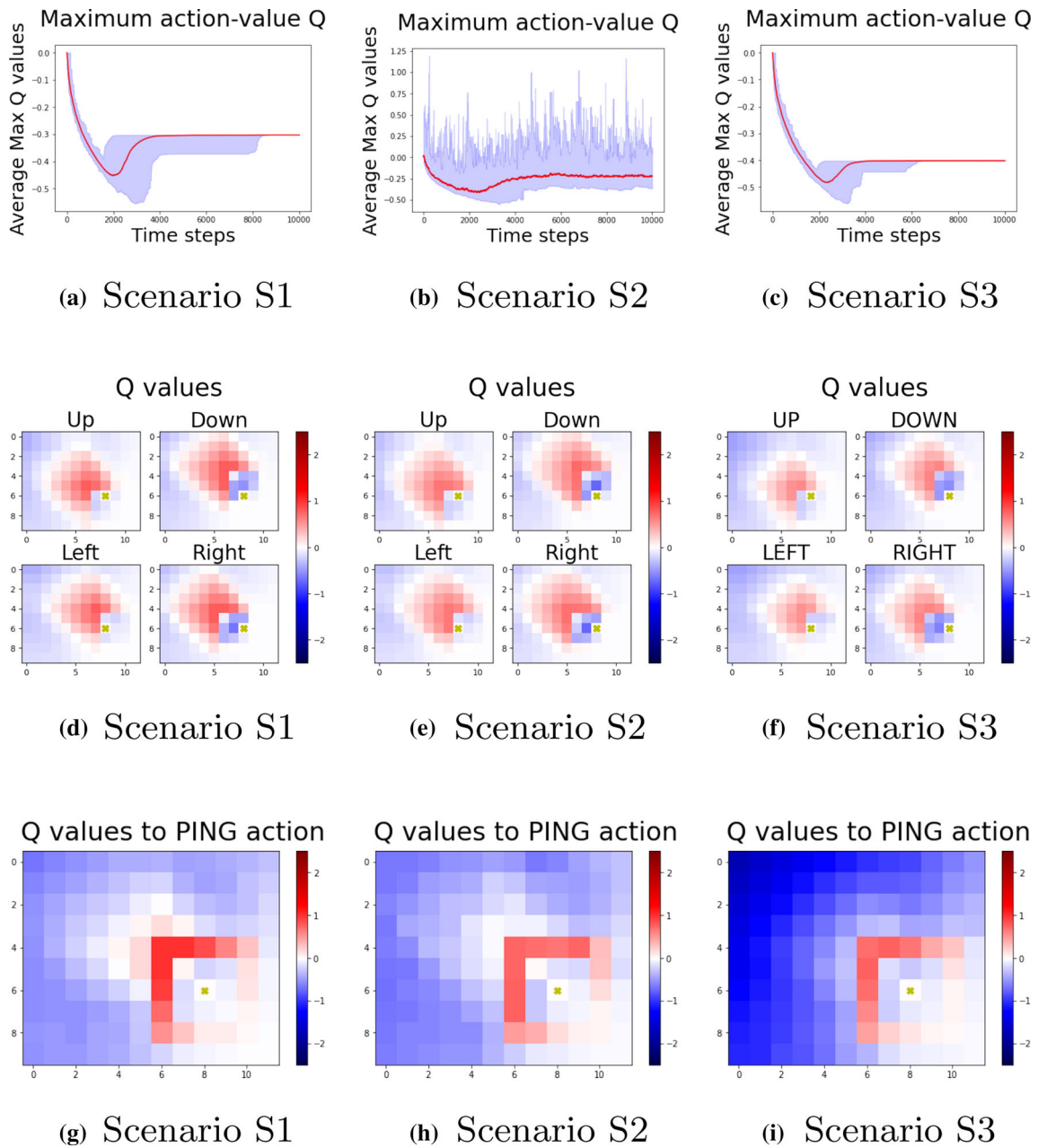


Fig. 7 Results of our experiments in three scenarios. The first row (a–c) shows the average of maximum Q-values per time step, the shaded area is between the maximum and minimum Q values from 100

agents. The second row (d–f) shows the average of the final Q-value for the actions UP, DOWN, LEFT, and RIGHT. The last row (g–i) shows the average final Q-values for the PING action

one, the agent maintains a constant communication with the issuer until the task has finished.

5.2.2 Disagreement area

Finally, we study the behavior of the learning agent in the robot approaching in disagreement area problem. Figure 11 shows the average discounted reward for 20 agents. We can observe that until 4000 episodes the average reward is increasing. However, in episode 4000, there is a

breakpoint, where the average reward decreases but has a growing tendency afterward. This behavior suggests that the weights of the value function $Q(x_t)$ could reach a different local minimum in these episodes.

Figure 12 shows the trajectories of the agent with highest reward in the validation phase. These are shown only when the agent selects the STOP action. Due to the start position, the agent does not complete the task successfully in some parts of the coordinate plane. We also observe that with angles $-\pi$ and $-\frac{3}{4}\pi$, the agent completes

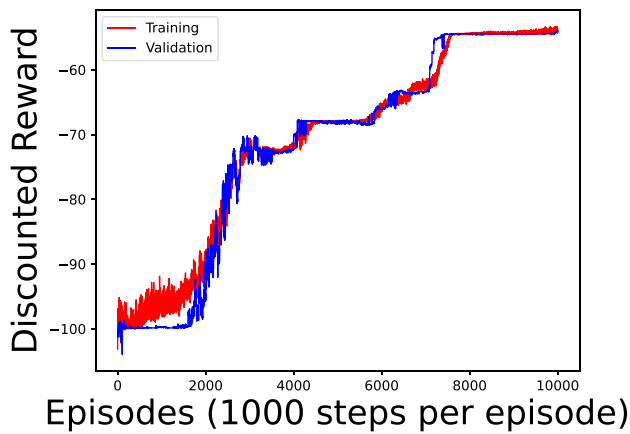


Fig. 8 Average collected discounted reward over 20 agents using A2C in the robot approaching in uncomfortable region environment. The red line is the collected reward during the training, and the blue line is the collected reward during the validation phase

fewer successfully trajectories in comparison with other angles. Another important aspect is that several trajectories do not even reach the frontier of the disagreement area. This performance can be explained by the STOP action. Figure 13 shows the trajectories with the probability of STOP. Similarly to the PING action, the STOP action divides the coordinate plane into two big areas. In the first one, the agent is free to move, and in the second one, the agent is obligated to stop. The light red region is an overestimate of disagreement areas. We empirically mark

this part of the field. In that section of the coordinate plane, the agent decides to stop because of the high negative reward, manifesting a conservative behavior by the agent.

5.3 Discussion

In the environments presented above, the agent can complete the task successfully. However, there is a dependency on the agent's initial position that affects its performance in areas far from that point. Also, the fixed position of the issuer is an essential factor for agent learning, because the agent could not perform well if the issuer moves through space. On the other hand, the agent can identify the best path to reach the region of interest, obtaining the highest reward. Furthermore, the agent can identify those regions where it can freely move without suffering punishment (based on reward). In the grid-world problem, the Q-value of the PING highlights those points with the highest reward, which are also the points in the target region. In continuous environments, with the help of PING and STOP, we can see a division of the space that separates areas where it is more likely to select these actions. These regions are conservative because they move away from the target region, implying that the agent prefers to stop rather than invade that region. This situation is more evident in the disagreement area, where the STOP high probability area contains the disagree area (see Fig. 13, light red covers up the circle), the latter being smaller than the

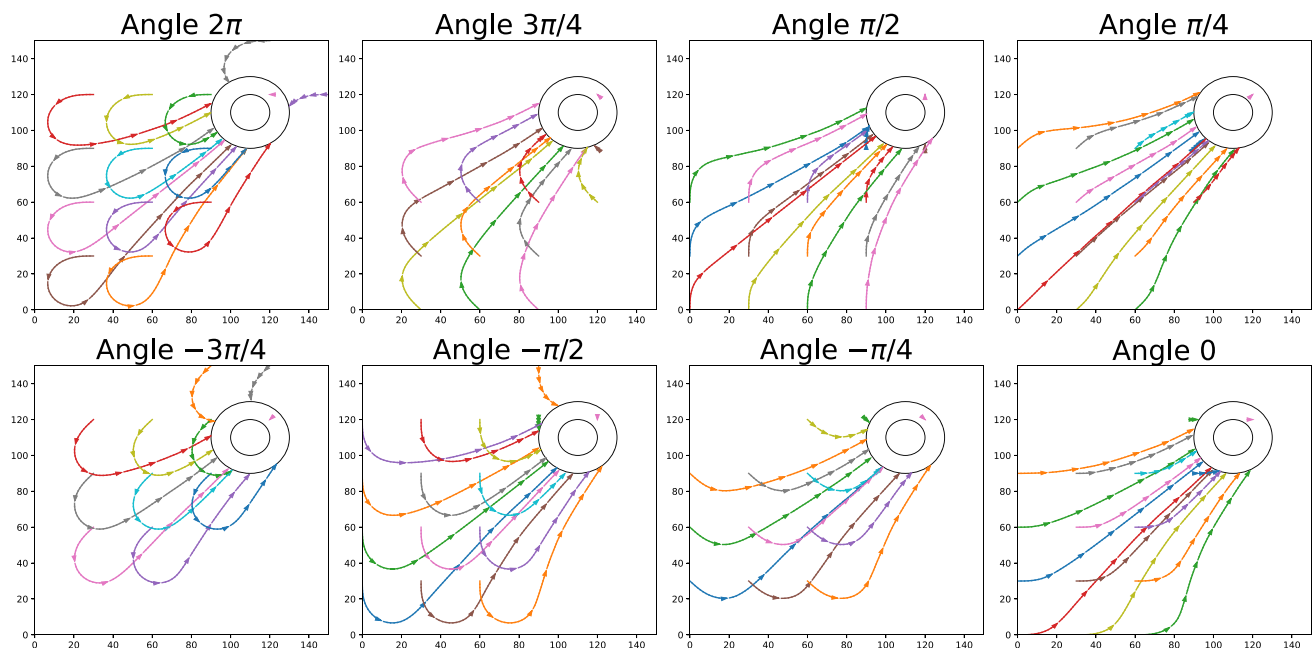


Fig. 9 Trajectories in the robot approach in uncomfortable region environment. The trajectories are computed from agent with the highest reward in the validation phase in different start positions. Only success trajectories are shown

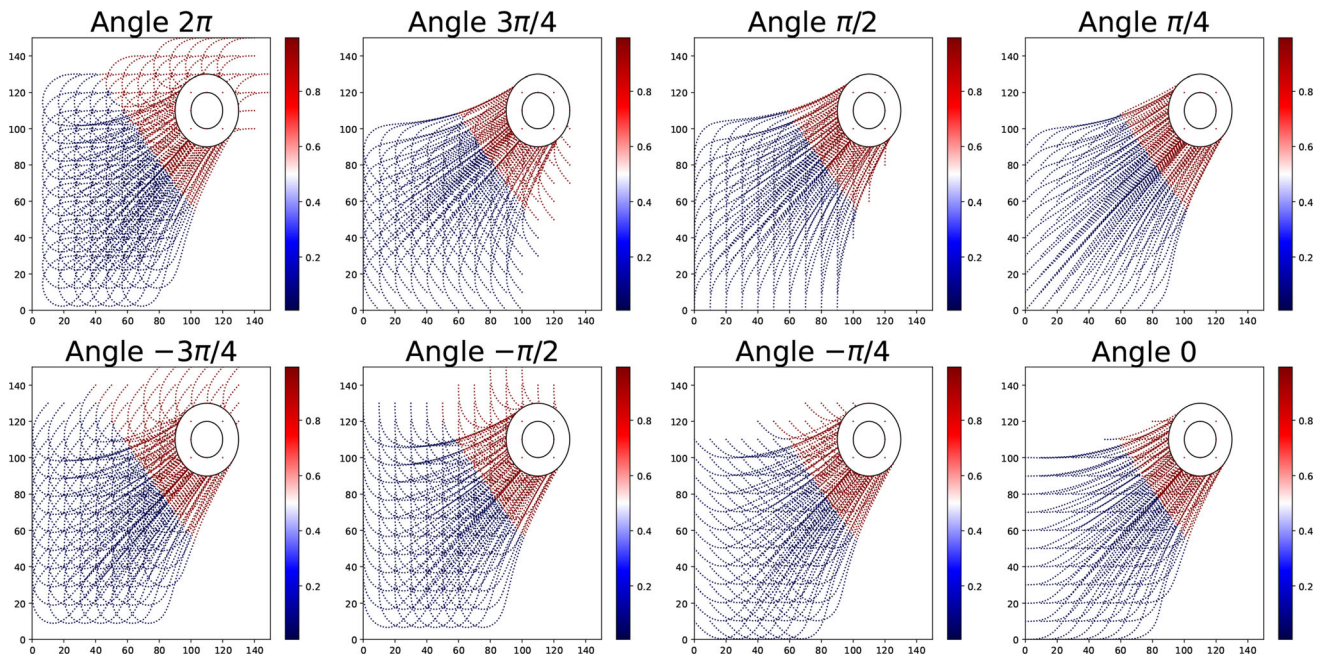


Fig. 10 Probability of sending PING in the robot approach in uncomfortable region environment. Red points show probability closer to one, and blue points show values closer to zero. The

trajectories are computed from agent with the highest reward in the validation phase from different start positions. Only success trajectories are shown

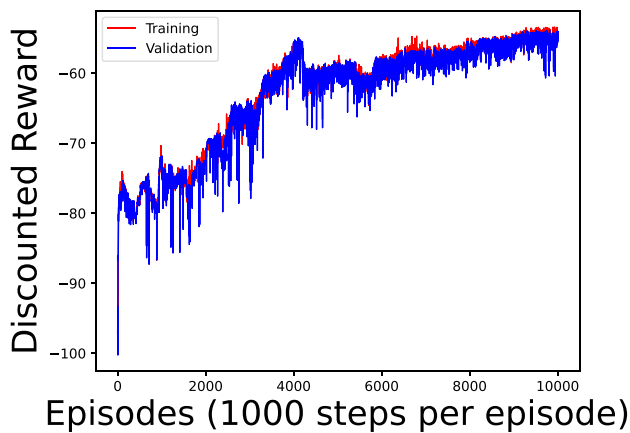


Fig. 11 Average collected discounted reward over 20 agents using A2C in the robot approaching in disagreement area environment. The red line is the collected reward during the training, and the blue line is the collected reward during the validation phase

former. Unlike the uncomfortable area, where the area of the high probability of sending PING is larger than the disagreement area. However, this is not surprising since the agent sends PING with information communicated by the issuer and not with the information perceived by the agent (in the disagreement area, the agent has information on the level of disagreement in its state).

6 Conclusions and future works

In this paper, we studied the agent performance in an environment based on proxemic behavior. We implemented a modified grid-world problem and the robot approaching problem in two versions, uncomfortable region, and disagreement area. These environments consider that an issuer agent gives information about its disagreement level to an RL agent that performs a trajectory, reaching the goal. Different regions around the issuer mimic their personal space, which replaces the traditional target of the agent. We implemented a Q-learning algorithm for the grid-world problem and A2C for the robot approaching problems. Our results show that the agent can reach the target region or get close to the issuer, even without giving information. Regarding the PING and STOP action, the agent manages to select these actions in a right place of the space, allowing the task to be performed satisfactorily. However, the agent is conservative because it performs these actions where the reward is not so high. On the other hand, the agent can identify nonconformity regions, even when it does not have information about them. This aspect gives the possibility to overestimate the personal space of the issuer (due to its conservative characteristic). The environments represent simply the interaction of the agent and the issuer, which helps to explore the behavior of the agent in an environment with proxemic behavior. However, it is necessary to include other features to approximate an environment with proxemic behavior

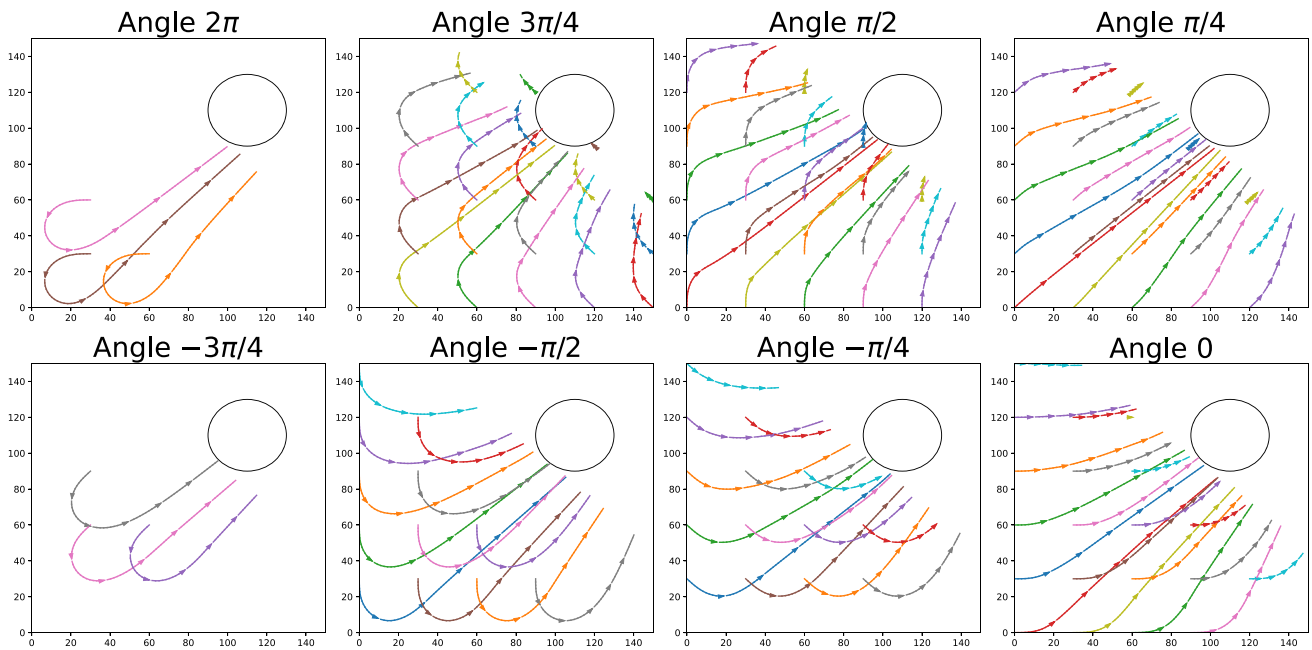


Fig. 12 Trajectories in the robot approach in disagreement area environment. The trajectories are computed from the agent with the highest reward in the validation phase from different start positions. Only success trajectories are shown

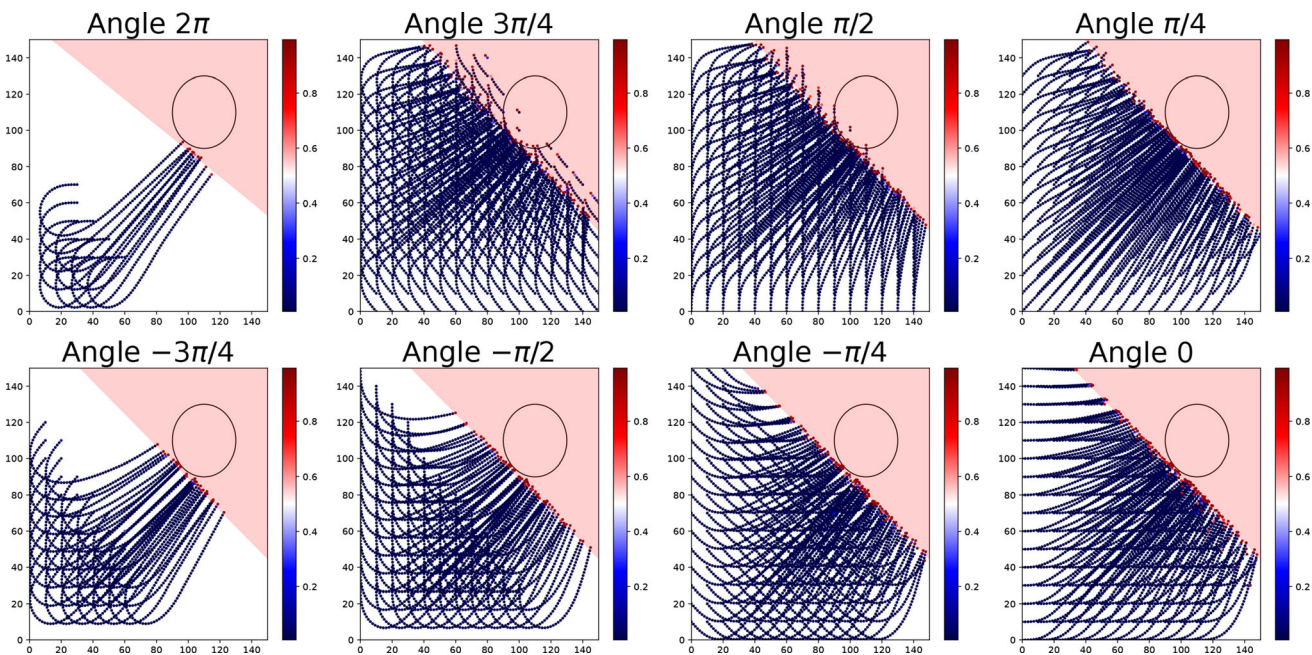


Fig. 13 Probability of STOP in the robot approach in disagreement area environment. Red points show probability closer to one, and blue points show values closer to zero. The light red region shows an

overestimate of the disagreement area. The trajectories are computed from the agent with the highest reward in the validation phase from different start positions. Only success trajectories are shown

accurately. For example, it is necessary for the personal space to be variable and to depend on characteristics expressed by the issuer, in addition to the agent positively or negatively influencing the issuer to modify their proxemics.

In future alignments, we explore identifying the proxemics region based on issuer information, such as gaze, orientation, emotion, or other aspects. Also, we intend to implement asymmetric proxemic regions to mimic human proxemic behavior. In our experiments, the fixed start position influences the trajectory performance and the

agent's capability to reach the target. Then, we intend to implement no fixed proxemic region, considering that the proxemic space changes by external factors. Finally, we intend to study the agent performance in more complex environments, involving other algorithms and techniques of reinforcement learning and deep learning.

Acknowledgements We would like to gratefully acknowledge financing in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES) - Finance Code 001, the Brazilian agencies FACEPE and CNPq - Code 432818/2018-9, and Universidad Central de Chile under the research project CIP2020013.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Hall ET, Birdwhistell RL, Bock B, Bohannon P, Diebold AR Jr, Durbin M, Edmonson MS, Fischer J, Hymes D, Kimball ST et al (1968) Proxemics [and comments and replies]. *Curr Anthropol* 9(2/3):83–108
- Zacharakis A, Kostavelis I, Gasteratos A, Dokas I (2020) Safety bounds in human robot interaction: a survey. *Saf Sci* 127:104667
- Churamani N, Cruz F, Griffiths S, Barros P (2020) icub: learning emotion expressions using human reward. [arXiv:2003.13483](https://arxiv.org/abs/2003.13483)
- Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press, Cambridge
- Millán C, Fernandes B, Cruz F (2019) Human feedback in continuous actor-critic reinforcement learning. In: Proceedings European symposium on artificial neural networks, computational intelligence and machine learning, Bruges (Belgium), pp 661–666
- Millán-Arias C, Fernandes B, Cruz F, Dazeley R, Fernandes S (2021) A robust approach for continuous interactive actor-critic algorithms. *IEEE Access* 9:104242–104260. <https://doi.org/10.1109/access.2021.3099071>
- Mumm J, Mutlu B (2011) Human-robot proxemics: physical and psychological distancing in human-robot interaction. In: Proceedings of the 6th international conference on human-robot interaction, pp 331–338
- Eresha G, Häring M, Endrass B, André E, Obaid M (2013) Investigating the influence of culture on proxemic behaviors for humanoid robots. In: 2013 IEEE Ro-Man, pp 430–435. IEEE
- Patompak P, Jeong S, Nilkhamhang I, Chong NY (2020) Learning proxemics for personalized human-robot social interaction. *Int J Soc Robot* 12(1):267–280
- Mitsunaga N, Smith C, Kanda T, Ishiguro H, Hagita N (2006) Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning. *J Robot Soc Jpn* 24(7):820–829
- Hediger H (1955) Studies of the Psychology and Behavior of Captive Animals in Zoos and Circuses. Criterion Books, Inc.
- Gunawan AB, Pratama B, Sarwono R (2021) Digital proxemics approach in cyber space analysis-a systematic literature review. *ICIC Express Lett* 15(2):201–208
- Lee M, Bruder G, Höllerer T, Welch G (2018) Effects of unaugmented periphery and vibrotactile feedback on proxemics with virtual humans in ar. *IEEE Trans Visual Comput Graphics* 24(4):1525–1534
- Li R, van Almkerk M, van Waveren S, Carter E, Leite I (2019) Comparing human-robot proxemics between virtual reality and the real world. In: 2019 14th ACM/IEEE international conference on human-robot interaction (HRI), pp 431–439. IEEE
- Sanz FA, Olivier A-H, Bruder G, Pettré J, Lécuyer A (2015) Virtual proxemics: Locomotion in the presence of obstacles in large immersive projection environments. In: 2015 IEEE virtual reality (vr), pp. 75–80. IEEE
- Llobera J, Spanlang B, Ruffini G, Slater M (2010) Proxemics with multiple dynamic characters in an immersive virtual environment. *ACM Trans Appl Percept (TAP)* 8(1):1–12
- Kastanis I, Slater M (2012) Reinforcement learning utilizes proxemics: an avatar learns to manipulate the position of people in immersive virtual reality. *ACM Trans Appl Percept (TAP)* 9(1):1–15
- Millán-Arias C, Fernandes B, Cruz F (2021) Learning proxemic behavior using reinforcement learning with cognitive agents. [arXiv:2108.03730](https://arxiv.org/abs/2108.03730)
- Ponce-López V, Escalera S, Baró X (2013) Multi-modal social signal analysis for predicting agreement in conversation settings. In: Proceedings of the 15th ACM on international conference on multimodal interaction, pp 495–502
- Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H (2018) State-of-the-art in artificial neural network applications: a survey. *Heliyon* 4(11):00938
- Abiodun OI, Jantan A, Omolara AE, Dada KV, Umar AM, Linus OU, Arshad H, Kazaure AA, Gana U, Kiru MU (2019) Comprehensive review of artificial neural network applications to pattern recognition. *IEEE Access* 7:158820–158846
- Lee JJ, Sha F, Breazeal C (2019) A bayesian theory of mind approach to nonverbal communication. In: 2019 14th ACM/IEEE international conference on human-robot interaction (HRI), pp 487–496. IEEE
- Kosiński T, Obaid M, Woźniak PW, Fjeld M, Kucharski J (2016) A fuzzy data-based model for human-robot proxemics. In: 2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN), pp 335–340. IEEE
- Gao Y, Wallkötter S, Obaid M, Castellano G (2018) Investigating deep learning approaches for human-robot proxemics. In: 2018 27th IEEE international symposium on robot and human interactive communication (RO-MAN), pp 1093–1098. IEEE
- Seker M, Männistö A, Iosifidis A, Raitoharju J (2021) Automatic social distance estimation from images: performance evaluation, test benchmark, and algorithm. [arXiv:2103.06759](https://arxiv.org/abs/2103.06759)
- Bochkovskiy A, Wang C-Y, Liao H-YM (2020) Yolov4: optimal speed and accuracy of object detection. [arXiv:2004.10934](https://arxiv.org/abs/2004.10934)

27. Cao Z, Simon T, Wei S-E, Sheikh Y (2017) Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7291–7299
28. Sun S, Zhao X, Li Q, Tan M (2020) Inverse reinforcement learning-based time-dependent a* planner for human-aware robot navigation with local vision. *Adv Robot* 34(13):888–901
29. Ramon-Vigo R, Perez-Higueras N, Caballero F, Merino L (2014) Transferring human navigation behaviors into a robot local planner. In: The 23rd IEEE international symposium on robot and human interactive communication, pp 774–779. IEEE
30. Luber M, Spinello L, Silva J, Arras KO (2012) Socially-aware robot navigation: a learning approach. In: 2012 IEEE/RSJ international conference on intelligent robots and systems, pp 902–907. IEEE
31. Charalampous K, Kostavelis I, Gasteratos A (2016) Robot navigation in large-scale social maps: an action recognition approach. *Expert Syst Appl* 66:261–273
32. Feldmaier J, Stimpfl M, Diepold K (2017) Development of an emotion-competent slam agent. In: Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pp. 1–9
33. Russell JA (1980) A circumplex model of affect. *J Pers Soc Psychol* 39(6):1161
34. Fuse Y, Takenouchi H, Tokumaru M (2021) Evaluation of robotic navigation model considering group norms of personal space in human–robot communities. In: *Soft computing for biomedical applications and related topics*, pp 117–125. Springer, Berlin
35. da Silva RR, Romero RAF (2011) Relational reinforcement learning and recurrent neural network with state classification to solve joint attention. In: The 2011 international joint conference on neural networks, pp 1222–1229. IEEE
36. Silva R, Policastro CA, Zuliani G, Pizzolato E, Romero RA (2008) Concept learning by human tutelage for social robots. *Learn Nonlinear Models* 6(4):44–67
37. Lin LJ (1991) Programming robots using reinforcement learning and teaching. In: AAI-91 the ninth national conference on artificial intelligence, pp 781–786. <http://www.aaai.org/Library/AAAI/1991/aaai91-122.php>
38. Lim M-H, Ong Y-S, Zhang J, Sanderson AC, Seiffert J, Wunsch DC (2012) Reinforcement learning. *Adapt Learn Optim* 12:973–978. <https://doi.org/10.1007/978-3-642-27645-3>
39. Bellman RE (2003) *Dynamic programming*. Dover Publications, New York
40. Watkins CJ, Dayan P (1992) Q-learning. *Mach Learn* 8(3–4):279–292
41. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In: *International conference on machine learning*, pp 1928–1937. PMLR
42. Grondman I, Vaandrager M, Busoniu L, Babuška R, Schuitema E (2012) Efficient model learning methods for actor-critic control. *IEEE Trans Syst Man Cybern Part B (Cybern)* 42(3):591–602. <https://doi.org/10.1109/TSMCB.2011.2170565>
43. Sutton RS, McAllester D, Singh S, Mansour Y (1999) Policy Gradient Methods for Reinforcement Learning with Function Approximation. In: *Proceedings of the 12th international conference on neural information processing systems*, pp 1057–1063. MIT Press Cambridge, Denver, CO
44. Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M (2014) Deterministic policy gradient algorithms. In: *International conference on machine learning*. ICML, Beijing. <http://proceedings.mlr.press/v32/silver14.pdf>
45. Baird LC (1994) Reinforcement learning in continuous time: advantage updating. In: *Proceedings of 1994 IEEE international conference on neural networks (ICNN'94)*, vol 4, pp 2448–2453. IEEE. <https://doi.org/10.1109/ICNN.1994.374604>. <http://ieeexplore.ieee.org/document/374604/>
46. Grondman I (2015) Online model learning algorithms for actor-critic control. PhD thesis, Delft University of Technology. <https://doi.org/10.4233/uuid:415e14fd-0b1b-4e18-8974-5ad61f7fe280>
47. Bhatnagar S, Sutton RS, Ghavamzadeh M, Lee M (2009) Natural actor-critic algorithms. *Automatica* 45(11):2471–2482. <https://doi.org/10.1016/j.automatica.2009.07.008>
48. Miller WT, Sutton RS, Werbos PJ (1995) *Neural networks for control*. MIT Press, Cambridge
49. Ghavamzadeh M, Mahadevan S (2003) Hierarchical policy gradient algorithms. *Computer Science Department Faculty Publication Series*, 173
50. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp 249–256. JMLR Workshop and Conference Proceedings

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.