MULTI-AGENT DEEP REINFORCEMENT LEARNING IN ROBOTICS: CONTEXT AND OPEN CHALLENGES

Maxime Toquebiau[*]
Vrije Universiteit Brussel, Artificial Intelligence Research Group, Belgium
maxime.toquebiau@vub.be

Jae-Yun Jun
ECE Paris

Faïz Benamar
Sorbonne Université, CNRS, ISIR, F-75005 Paris, France

Nicolas Bredeche
Sorbonne Université, CNRS, ISIR, F-75005 Paris, France

ABSTRACT

Real-world environments frequently involve multiple interacting entities whose behaviours co-evolve, making learning and decision-making fundamentally more complex than in the single-agent setting. Multi-agent deep reinforcement learning (MADRL) addresses such scenarios by developing new techniques inspired from multi-agent systems and deep reinforcement learning. One objective of this research to tackle robotic settings, where teams of robots must coordinate, cooperate, or compete under real-world constraints. Yet, despite progress on algorithms, several gaps remain between MADRL research and practical robotic deployment. This survey provides a structured overview of MADRL from the perspective of robotic applications. We first formalise key concepts of multi-agent learning and challenges introduced by decentralisation, non-stationarity, partial observability, and coordination. We then analyse how these challenges intersect with robotics-specific considerations such as embodiment, safety, and sim-to-real transfer. Following this, we survey the MADRL literature, outlining the main research directions in the field from the last decade. Finally, we reflect on the shortcomings of current approaches and identify avenues for advancing cooperative MADRL towards scalable, robust, and deployable multi-robot intelligence.

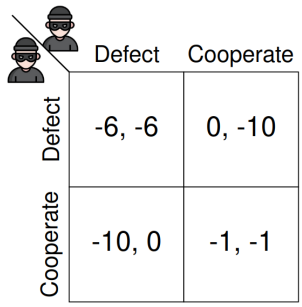
Contents

[*]Author was affiliated to *Sorbonne Université*, Paris, and ECE Paris when producing this work.

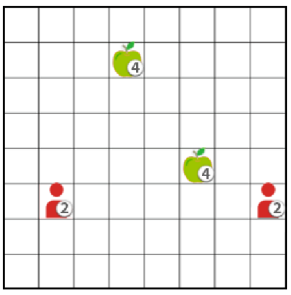
1 Introduction

In most realistic environments, multiple entities interact with each other to fulfil an individual or a collective goal. With multiple entities, each with their personal reasoning, the outcome of one's actions also depends on the others' actions, thus increasing the difficulty of learning how to best behave. If entities are all capable of adapting and learning, then the environment becomes an ever-evolving sum of intersecting strategies. The problem of finding the optimal strategy becomes even more complex, as the best response to previously observed behaviours might not be true in future tries. The single-agent learning setting does not explicitly model these new problems and, thus, falls short in most of these cases. For this reason, the concept of multi-agent system (MAS) has been defined to better describe the dynamics of environments containing multiple intelligent entities, as found in human societies (Doran & Palmer, 1995; Bousquet & Le Page, 2004; Hamill & Gilbert, 2015), games (Nowé et al., 2012; Owen, 2013), or robotics (Parker et al., 2016; Rizk et al., 2019) (see Figure 1 for example of multi-agent environments). Multi-agent learning specifically tackles how learning takes place in MASs, how it can be harmed by having multiple agents, and how it can benefit from it. Multi-agent learning research comes from the intersection of many different views in software engineering (Ben-Ari, 2006), distributed artificial intelligence (Stone & Veloso, 2000), and game theory (Rosenschein & Zlotkin, 1994; Leyton-Brown & Shoham, 2008); each one having proposed ways of modelling multi-agent interactions (Wooldridge, 2009). One possible way is to adapt and extend single-agent reinforcement learning (RL) tools to fit the needs of MASs. Recent years have seen the development of multi-agent deep reinforcement learning (MADRL) algorithms, with a wide range of new approaches for tackling richer multi-agent environments.

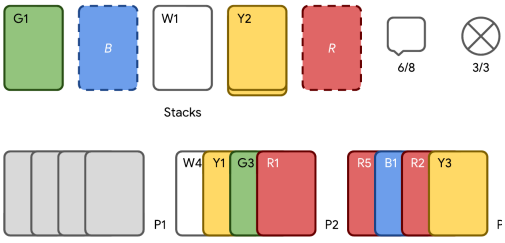
Many robotic applications involve interactions between multiple robots and/or humans (Parker et al., 2016). In this context, MADRL is a potentially valuable tool for learning complex multi-agent behaviours in realistic environments (Orr & Dutta, 2023). But, to progress towards this objective, we need to ask ourselves: *What does it mean to have robots*



(a) Social dilemma

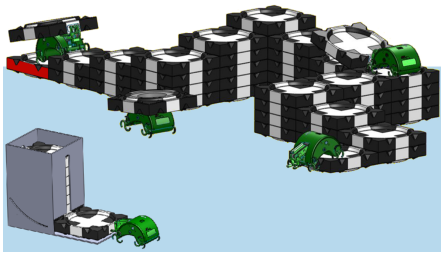


(b) Two-dimensional grid world

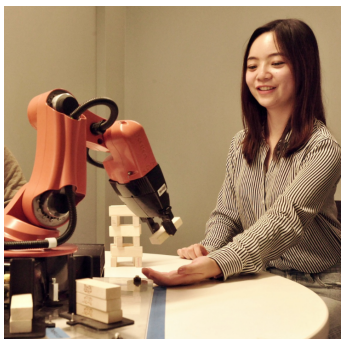


(c) Card game

(d) Multi-player video game



(e) Multi-robot system



(f) Human-robot interaction

Figure 1: Types of multi-agent environments. (a) Social dilemmas, here the famous *prisoners' dilemma*, extensively used in the game theory literature to devise multi-agent learning concepts and exact solutions. (b) Two-dimensional grid environments (here, level-based foraging; Albrecht and Ramamoorthy, 2013) that allow studying various multi-agent tasks in simplified settings. (c) Card and tabletop games (here, Hanabi; Bard et al., 2020) that often require learning complex strategies. (d) Multi-player video games (here, Dota 2, tackled by OpenAI et al., 2019) that offer rich environments demanding complex team-play. (e) Multi-robot systems (here, the TERMES construction robots; Petersen et al., 2012), having to deal with high-dimensional robotic settings. (f) Human-robot interaction, with the need to build interfaces and adapt to human partners (Jung et al., 2020).

operating in the real world? What should be the requirements when building intelligent robots and how should this impact the design of learning algorithms? This thesis intends to provide answers to these questions by studying existing approaches and proposing new solutions for improving cooperative MADRL in the context of robotics. In this particular regard, we are faced with several observations about the related literature:

1. Learning with multiple agents implies multiple theoretical and technical issues that are often treated separately or even ignored.

2. Similarly, robotic domains establish several constraints and challenges that are not thoroughly investigated in MADRL research, despite a prevalent aspiration of applying these algorithms to robotic settings.

3. The multi-agent (deep) reinforcement learning literature is extremely rich, with many orthogonal subjects of interest. This results in a domain that is hard to grasp and fully understand, with many works that are difficult to compare and evaluate.

This article aims to propose a clarified view of the literature, providing insights and avenues for reflection on these three starting observations. In Section 2, we start by formally defining important concepts and mathematical tools used in multi-agent RL. In Section 3, we introduce points (1) and (2), defining the challenges met in multi-agent learning and robotic settings, and looking at how they can overlap. To answer point (3), in Section 4, we present a short review of recent works in the MADRL literature. Finally, in Section 5, we provide a personal reflection on the remaining shortfalls of MADRL research, specifically when dealing with robotic applications. We aim to highlight the major flaws of this line of work and identify potential areas of improvement.

2 Multi-Agent Learning: Definitions

Multi-agent learning studies how learning can take place in environments where there are two or more intelligent agents. There is no theoretical limit to the number of agents in a MAS, but studies in this field are often limited to rather small groups of agents (say, from 2 to around 20), with larger groups being the subject of swarm robotics (Hamann, 2018). With a limited number of agents, multi-agent learning research can study more complex agent definitions and more intricate social dynamics.

Agents in a MAS are computer systems with the ability to observe and act upon an environment in which they are contained. They usually have a task to fulfil, which will require to learn a strategy of actions in the environment that satisfies the requirements of the task. Because multiple of these agents are present in the system, they can be considered as *social* entities, with specific social abilities, which can be either *explicit abilities*: e.g., communication (see Sections 2.3 and 4.5), prediction of multi-agent outcomes (see Section 2.4), or agent modelling (see Section 4.6); or *implicit abilities*: e.g., cooperation, coordination, attack/defence, negotiation, or bluffing. Taking this social aspect into consideration will influence the design of multi-agent learning algorithms.

2.1 Learning Framework: Dec-POMDP

Because we want to study cooperation in robotic-like environments, we use the decentralized partially-observable Markov decision process (Dec-POMDP) (Oliehoek & Amato, 2016) as our learning framework. The Dec-POMDP is an extension of the single-agent MDP that allows the presence of multiple agents and models the fact that the environment is not fully observable: agents only observe a part of it through their sensors. Formally, it is defined as a tuple $\langle \mathbf{S}, \mathbf{A}, \mathcal{T}, \mathbf{O}, \mathcal{O}, \mathcal{R}, n, \gamma \rangle$ in which:

- n is the number of agents;
- $\mathbf{S}$ is the set of all possible states of the environment, often referred to as "global states" as they describe the environment entirely;
- $\mathbf{O}$ is the set of joint observations, with one joint observation $\mathbf{o} = \{o^1, ..., o^n\} \in \mathbf{O}$ being a set of local observations (one for each agent);
- $\mathbf{A}$ is the set of joint actions, with one joint action $\mathbf{a} = \{a^1, ..., a^n\} \in \mathbf{A}$ being a set of local actions (one for each agent);
- $\mathcal{T}$ is the transition function defining the probability $P(s'|s, \mathbf{a})$ to transition from state s to next state s' with the joint action $\mathbf{a}$;
- $\mathcal{O}$ is the observation function defining the probability $P(\mathbf{o}|\mathbf{a}, s')$ to observe the joint observation $\mathbf{o}$ after taking joint action $\mathbf{a}$ and ending up in s';
- $\mathcal{R} : \mathbf{S} \times \mathbf{A} \to \mathbb{R}$ is the reward function producing a single reward for all agents at each time steps;
- $\gamma \in [0, 1)$ is the discount factor controlling the importance of immediate rewards against future gains.

Figure 2 illustrates a time step in the Dec-POMDP framework. An important thing modelled in this framework is that agents may not have access to the global state of the environment s_t (it is often even not defined). They only observe a sub-part of this global state through their sensors, which is represented by the local observations o^i_t. In a robotic environment, the local observation will be the result of the robot's sensors (e.g., camera, lidar, etc.). In simulation, the content of the observations is typically defined arbitrarily by deciding what an agent should be able to observe from the environment (e.g., its position, the relative positions of other agents, etc.). The joint observation $\mathbf{o}_t$, denoted in bold, is the concatenation of all local observations at step t (and similarly for the joint action $\mathbf{a}_t$).

2.2 Multi-Agent Reinforcement Learning Tools

The Dec-POMDP framework allows the development of RL agents, with the basic RL tools adapted to take into consideration the multi-agent context. With multiple agents, the goal of the learning algorithm is to find the optimal **joint policy** $\pi = (\pi^1, ..., \pi^n)$, with one **local policy** π_i for each agent i, that maximises expected future returns. In the Dec-POMDP, local policies are conditioned on the action-observation history $h^i_t = (o^i_0, a^i_0, ..., a^i_{t-1}, o^i_t) \in \mathbf{H} = (\mathbf{O} \times \mathbf{A})^*$ that contains all previous local observations and actions in the current episode. Thus, we have one local policies $\pi^i(a^i|h^i) : \mathbf{H} \times \mathbf{A} \to [0, 1]$ for each agent. The **joint value functions**, related to the joint policy can be written as:

$$V_\pi(\mathbf{h}_t) \coloneqq \mathbb{E}_\pi[G_t \mid \mathbf{h}_t] \text{ and } Q_\pi(\mathbf{h}_t, \mathbf{a}_t) \coloneqq \mathbb{E}_\pi[G_t \mid \mathbf{h}_t, \mathbf{a}_t]; \tag{1}$$

and similarly for the **local value functions** related to the local policy π_i:

$$V_{\pi^i}(h^i_t) \coloneqq \mathbb{E}_{\pi^i}[G_t \mid h^i_t] \text{ and } Q_{\pi^i}(h^i_t, a^i_t) \coloneqq \mathbb{E}_{\pi^i}[G_t \mid h^i_t, a^i_t], \tag{2}$$

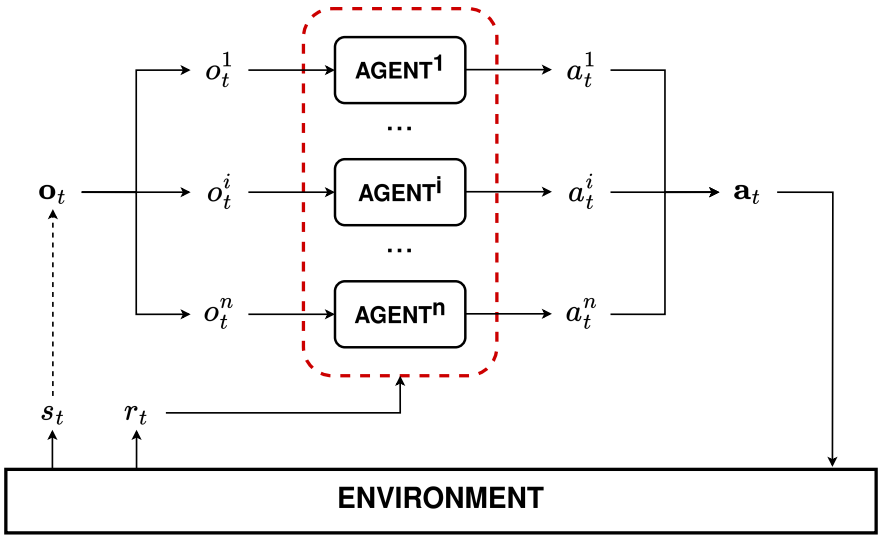


Figure 2: Diagram illustrating the Dec-POMDP framework. At each step t, the environment is in a state s_t that may be unknown by the agents. The joint observation $\mathbf{o}_t$ is produced by the environment, containing one local observation o_t^i for each agent i. Each agent produces an action a_t^i, all actions being gathered in the joint action $\mathbf{a}_t$ that is to be executed in the environment. A single reward r_t is produced and shared by all agents for evaluating step $t-1$. The red dotted line symbolises the multi-agent learning algorithm that can use the reward in various ways to train the agents.

with the discounted return $G_t \coloneqq \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, $\gamma \in [0,1]$, and $\mathbf{h}_t = \{h_t^1, ..., h_t^n\}$ the joint action-observation history. We will also denote the local policy of any agent i as π_{θ_i}, with θ_i the learnt parameters of agent i. Similarly, the joint policy can be written π_θ, with the whole set of parameters $\theta = (\theta_1, ..., \theta_n)$.

Conditioning on the history rather that the observation is common in MADRL, although not systematic. This models the fact that agents may have some form of memory of what happened in previous time steps of the episode. Concretely, this often corresponds to using recurrent neural networks in the agent architecture.

2.3 Communication

Multi-agent interactions may benefit from the ability to communicate information to other agents. Communicating efficiently requires knowing what information should be shared and how to share this information to be understood correctly. Learning these skills is a challenge in itself that has been thoroughly studied in dedicated lines of work (Austin, 1975; Farrell & Rabin, 1996; Brighton et al., 2005; Galke et al., 2022) and in the context of MADRL (Zhu et al., 2024). In a MAS, communication can take multiple forms. Agents might develop implicit communication abilities, using physical actions to convey abstract information (e.g., pointing a finger towards an object). Agents might use a shared archive, similar to a blackboard in a room, to store information accessible to everyone. Or, they might be provided a collection of mechanisms for exchanging messages between agents. We focus on such explicit, message-based communication to describe how agents can learn to communicate.

Communication mechanisms can be defined in many different ways depending on when agents communicate, how they share information, and to whom they are allowed to communicate. Here, we define a formal framework for communication to describe its use in MADRL algorithms. Note that this framework might not fit all communication architectures perfectly, but it can be adapted if needed.

We define communication as taking place during the action-selection process of the agents. After receiving their local observations, agents take part in a *communication turn* where they can generate a message $m_t^i = f_{comm}^i(h_t^i, \theta_i)$ and send it to the other agents. We consider f_{comm}^i as a learnt module of agent i, using a subset of θ_i (e.g., a dedicated neural network) to generate the message. The message can be sent to either all other agents, referred to as *broadcasting*, or to a limited subset, e.g., within a certain range or choosing particular agents to target. Agents then receive incoming messages and use them according to their architecture. Some methods might allow repeating such communication turn multiple times to allow some form of discussion. Then, the incoming information is used to compute the generated action: $a_t^i \sim \pi_i(.|h_t^i, m_t)$, where m_t refers to the incoming messages.

2.4 Nash Equilibrium

An important concept from game theory is the Nash equilibrium, introduced by Nash, 1950. It represents a stable joint policy state where no player can gain an advantage by changing their individual strategy, provided that the other players' strategies remain unchanged. This equilibrium can take various forms. A deterministic Nash equilibrium requires agents to always choose a particular joint action. A stochastic Nash equilibrium is one describing a stochastic joint policy, where agents select their actions given a specific equilibrium distribution. Nash equilibria are pivotal in multi-agent systems research, as they help predict the behaviour of agents interacting in competitive or cooperative environments.

Importantly, there may be multiple Nash equilibria in a particular environment, with some equilibria yielding better returns than others. Thus, a problem of equilibrium selection arises to avoid suboptimal equilibria and efficiently converge to the optimal one (Harsanyi & Selten, 1988; Kalai & Lehrer, 1993; Bowling & Veloso, 2002; Conitzer & Sandholm, 2007). While equilibria are instructive for understanding valuable long-term strategies, their direct application may be flawed in dynamic and unpredictable environments (Shoham et al., 2007). Additionally, they are impractical, if not impossible to compute in high-dimensional and non-simulated environments, as they require knowing the outcomes of all actions in all possible states. Nonetheless, Nash equilibria remain a significant concept in multi-agent learning, helpful for illustrating the dynamics of some multi-agent scenarios.

3 Context: Challenges in Multi-Agent Robotic Domains

Working towards learning behavioural strategies for multi-robot systems requires first understanding the implications of learning in such environments. The multiplicity of intelligent agents in an environment has many important impacts on how these agents are able to learn, disrupting basic RL techniques and imposing strong architectural choices. Similarly, learning in robotics faces several obstacles that hinder the applicability of algorithms designed to learn in games and simulations (Pierson & Gashler, 2017). In this section, we define the challenges of both domains, look at how they intersect, mention some basic or foundational approaches to tackle them, and discuss their impact on the design of learning algorithms.

3.1 Learning with Multiple Agents

3.1.1 Non-Stationarity

The problem of non-stationarity refers to the continuously changing nature of the learning environment. It occurs in single-agent RL, as training the agent changes its behaviour, thus modifying the nature of its future interactions with the environment. But, in the multi-agent setting, non-stationarity is greatly amplified by having all agents change their behaviour after each training phase. From the point of view of an agent, the optimisation problem (i.e., "finding the best policy to maximise future returns") changes after each training phase, because the distribution of states and the outcome of actions change. This leads to a moving target problem, where continuous co-adaptation can induce unstable training. This is especially bad for RL algorithms, as they bootstrap the learnt estimates to determine the best solution to the optimisation problem. If this problem changes too much after each training phase, the learnt estimates can end up ineffective. In addition, the optimisation problem being constantly evolving, past experiences are made irrelevant for future training phases as they depict obsolete interactions. This makes the use of experience replay unusable without specific adaptation (Foerster et al., 2016a).

Non-stationarity is a major issue preventing single-agent RL algorithms from working in multi-agent settings. To tackle this problem, multiple approaches have been proposed. In Section 4, we will see that using centralised information during training can help constrain the effects of non-stationarity. Because experience replay is such an important part of making deep RL work, some works have proposed solutions to adapt it for MADRL by ensuring that all agents train on concurrent time steps (Omidshafiei et al., 2017) and giving less weight to older experiences in the replay memory (Foerster et al., 2017). While the concurrent replay method has been widely adopted, this does not completely solve the issue. Replay memories in MADRL are often set to discard old experiences earlier than in single-agent RL to prevent learning from too irrelevant data.

3.1.2 Credit Assignment

Credit assignment refers to the problem of finding how each past action contributed to the obtained rewards. As for non-stationarity, this problem exists in single-agent RL, with actions contributing to rewards obtained in the future, requiring a form of *temporal credit assignment*. But this becomes harder with multiple agents as, at each step, multiple actions are performed simultaneously by different agents. Knowing the marginal contribution of each action is a crucial requirement for evaluating each local policy accordingly. This **multi-agent credit assignment** problem can be

examined from the point of view of the whole MAS, looking at *"how much each agent contributed to the obtained return?"*, or from the point of view of a single agent, answering the question *"was my success (or failure) due to my actions, or to some other agent's actions?"*. This problem is particularly difficult in the Dec-POMDP framework where all agents share a common reward signal. In such cases, a "lazy" agent could be reinforced into doing nothing because the other agents manage to solve the task on their own. However, it is worth noting that the problem persists if each agent gets its own individual reward (and, similarly, in competitive settings), because a local reward might have been caused by another agent's actions.

To tackle the multi-agent credit assignment problem, some methods have been developed for trying to model the marginal contributions of each agent. Exact solutions exist in simple, controlled environments, like the *Shapley value* (Shapley, 1953) that evaluates an agent with the expected gain in utility of having this agent contribute in any possible coalition of agents. But this value is computationally very expensive and even approximations are hardly applicable to environments outside the game-theoretic framework or with a large number of agents (Fatima et al., 2008; Michalak et al., 2014; J. Wang et al., 2020, 2022). Another approach is to compute the marginal contribution of each action as the difference between the observed outcome and a counterfactual outcome where the evaluated action was not performed. The *wonderful life utility* (WLU; Wolpert and Tumer, 1999) replaces the action by a "null" action and simulates the presumed return: $WLU(a_i) = G(\mathbf{a}) - G(\mathbf{a}_{-i}, a_i = \text{null})$, with G the evaluation function and $\mathbf{a}_{-i}$ the joint action without action a_i. The *aristocrat utility* (AU; Wolpert and Tumer, 2002) takes as counterfactual metric the expected outcome from the truncated joint action: $AU(a_i) = G(\mathbf{a}) - \mathbb{E}(G|\mathbf{a}_{-i}, s)$, with s being the state in which action a_i was performed. However, these utilities are impractical to compute in high-dimensional, stochastic environments. In Sections 4.3 and 4.4, we will introduce recent approaches, based on MADRL, to the credit assignment problem that better fit more realistic settings.

3.1.3 Coordination

In many cooperative multi-agent tasks, agents are required to coordinate their actions to fulfil the objective. Robots might have to lift a heavy object together. In games played in teams, coordination between teammates is often crucial to successful tactics (Samvelyan et al., 2019; Bard et al., 2020). Thus, coordination is often a pursued ability in cooperative multi-agent learning. Coordination can be defined, in the multi-agent context, as the ability of an agent to synchronise its local actions with the anticipated behaviour of other agents to achieve a particular objective. This ability encompasses a wide set of knowledge the agents must acquire: knowledge about the environment dynamics, about how the task is completed, and about the other agents' strategies. Learning all this can be extremely complicated, especially in partially observable environments where local information is often insufficient to understand the full state of the environment.

Coordination skills may be learnt with different approaches. First, by acquiring a thorough knowledge of the joint policy search space to know how to behave in any situations. This can be related to a problem of multi-agent exploration of the joint policy space (see Section 5.2). Another approach would be to have an explicit mechanism for coordinating actions during execution. This can be tackled in various ways. In Section 4.6, we review methods that enable agents to learn a model of the other agents' policies, allowing them to coordinate their actions. Communication can be a handy mechanism for exchanging local information and reaching a consensus on the best way to act (see Section 4.5). Finally, coordination graphs (Guestrin et al., 2002; Böhmer et al., 2020; S. Li et al., 2021) more explicitly model coordination by learning pairwise joint value functions.

3.1.4 Scaling

A major issue of all multi-agent learning algorithms is how they deal with scaling to larger state and action spaces and, especially in the case of MASs, to a larger number of agents in the system. Being able to handle more agents efficiently makes an algorithm more applicable to various settings. But, this is not elementary, as scaling exacerbates all issues faced by multi-agent learning. Non-stationarity is increased because having more agents in the system means the environment changes more after each training step. Credit assignment is harder because the outcomes depend on more local actions. Learning value and policy functions is more difficult, as they depend on more independent elements and, thus, the search space is larger. On a more technical note, having more agents makes training more computationally expensive. As in RL, the use of deep learning techniques can help for dealing with scaling. It helps generalising to unseen configurations of the environment, partly compensating for the larger search spaces. In Section 4.2, we will see that having agents learn independently can help with scaling, but it also implies some important downsides.

3.2 Learning in Robotic Domains

3.2.1 Domain Complexity

Going towards learning in robotic environments, an important obstacle to successful learning is the complexity of realistic environments. Many multi-agent environments simplify the definition of states and actions substantially to focus on particular multi-agent dynamics (e.g., two-dimensional grid level-based foraging, as shown in Figure 1b, to study how coordination can arise). But, to apply learning algorithms to robotic domains, they must be capable of handling the full complexity of realistic state and action definitions. This complexity takes three different forms: high-dimensionality, continuity, and multi-modality.

High-dimensional states and actions are made of many components that each carry some information. Take, for example, a robotic hand manipulating objects. For observing its environment, it might use an RGB (red-green-blue) camera producing images made of thousands of three-valued pixels (one value for each colour). Understanding each image requires knowing how these numerous values relate with each other to compose high-level information. For manipulating objects, the robotic hand must control a large number of motors at once to achieve the desired motion. Each motor necessitates its own policy, but all policies must coordinate to achieve the desired behaviour. Note that, such high-dimensional action space can benefit from being formulated as a multi-agent problem, with each joint being handled by one agent (Sartoretti et al., 2019).

Robots usually deal with **continuous** inputs and outputs. For example, pixel values can be represented as a continuous value of colour intensity. Continuous angles of rotation are used to precisely articulate joints. Having continuous states and actions implies that the respective search spaces are infinite. This requires the ability to generalise well to handle previously unseen states: in a continuous state space, a robot will never experience the exact same state twice so it needs to use its experience in very similar states to know how to react. Thus, handling continuous states and actions requires specific design choices.

Finally, robots often encounter **multi-modal** states and actions, which are composed of multiple different types of information. In addition to the RGB camera, the robotic hand might have tactile sensors on each finger that generate precise haptic information on how the object is being grasped. Actions may also be distributed on different types of actuators: e.g., wheels to navigate in a room, hands to grasp objects, and voice to communicate. Multi-modality requires the controller to understand different types of information and recognise their relationships and interactions.

These input and output complexities each require specific care and altogether make learning more difficult. Deep learning techniques are an important tool for dealing with this level of complexity. They allow automatic learning of high-level representations from high-dimensional data (Simonyan & Zisserman, 2015), generalisation in continuous spaces (Schulman et al., 2016), and fusing of multi-modal data (Radford et al., 2021; Driess et al., 2023). With the right learning approach and enough resources, deep learning enables efficient learning of complex robotic behaviour (Pinto & Gupta, 2016; Andrychowicz et al., 2020). However, using deep learning has some notable drawbacks such as sample inefficiency and computational cost. When used in the context of RL, deep learning techniques are useful to open the range of potential applications, but RL techniques still require extensive work to suit robotic domains well (Sünderhauf et al., 2018; Ibarz et al., 2021).

3.2.2 Partial Observability

All realistic robotic environments are inherently partially observable. It is practically impossible to capture the full complexity of a real-world setting within a single state vector. A robot typically accesses information about its environment by observing it with its own sensors, providing it with a subjective view of its immediate surroundings. This limited perspective does not offer a complete description of the full state of the environment. To achieve its objective, a robot must infer some information based on its past experiences and observations from previous steps.

In the context of multi-agent RL, the Dec-POMDP, defined in Section 2.1, is useful for modelling this environmental uncertainty. Similar to realistic robotic settings, agents within a Dec-POMDP only observe a subjective subset of the complete state of the environment. The resulting uncertainty significantly increases the difficulty of learning optimal policies. To overcome this, robots require specific tools. In single-agent POMDPs, agents can learn to infer the current state of the environment from their incomplete observations (Abbeel et al., 2006; Lee et al., 2019). However, in a multi-agent setting, this is significantly harder as the state also depends on the actions of other agents (Papadimitriou & Tsitsiklis, 1987; Oliehoek & Amato, 2016). Memory can be a valuable tool in dealing with uncertainty, enabling agents to remember previous observations during an episode to better infer the current state (Hausknecht & Stone, 2015). Communication can also play a crucial role, allowing agents to share their individual subjective knowledge, thus cooperating to build a more accurate representation of the current state (see Section 4.5).

3.2.3 Reality Gap

Because RL requires thousands of experiences to converge to an efficient strategy, a promising approach to learning robotic tasks is to train in simulation and then apply the learnt policy on the real robots. However, this transfer is challenging due to numerous subtle differences between the simulated environment and the real world, a problem known as the **reality gap** (Jakobi et al., 1995). Even the most sophisticated simulations fall short of accurately reproducing real-world dynamics. Real robotic sensors and actuators have imperfections that are often not modelled in simulations. These minor discrepancies accumulate, causing a policy that performs well in simulation to fail when deployed on a real robot. To leverage extensive training in simulation, techniques must be developed to enable efficient simulation-to-reality (sim-to-real) transfer (Ju et al., 2022).

One possible sim-to-real approach is to develop models that adapt quickly and effectively to new domains (Rusu et al., 2017). This can be facilitated by learning the critical characteristics shared between the training and execution domains (A. Gupta et al., 2017; James et al., 2019). Another strategy, which can complement the first, is domain randomisation. This technique involves slightly randomising observations and actions during training in simulation, making RL agents more robust to minor discrepancies in their inputs and outputs (Tobin et al., 2017; Chebotar et al., 2019; Andrychowicz et al., 2020). Lastly, hierarchical learning offers a promising approach to developing transferable policies (Nachum et al., 2020; D'Ambrosio et al., 2024). Assuming that the reality gap affects lower-level actions more, it should be easier to learn a transferable high-level policy that selects macro-actions (e.g., find an apple) (Amato et al., 2019). Low-level policies for executing these macro-actions with basic actions (e.g., move forward) can be learnt more easily within the target domain.

3.2.4 Embodiment

Contrary to most computer programs, the controller of a robot is **embodied**: it is situated within a concrete body that lives inside an environment (Pfeifer & Bongard, 2006). There are many different views of embodiment from philosophy, psychology, cognitive science, and artificial intelligence, describing the role of embodiment in learning and intelligence (Lakoff & Johnson, 1999; Barsalou et al., 2003; Kiverstein, 2012; Sünderhauf et al., 2018). Here, we define some useful notions for appreciating the various implications of control and learning in robotics. **Physical** embodiment relates to the instantiation of the controller in a physical body that can sense and act upon its environment. **Temporal** embodiment refers to the fact that robots experience their environment through sequences of strongly correlated states. This has important implications for the treatment of both past states, on which the present state depends, and future states, which can be influenced by the robot's actions. Being embodied also entails a **compositeness** of the agent's body, made of many interconnected parts that all serve a specific purpose, for sensing or acting. Ziemke (2003) calls this "organismoid" to relate the composite body of robots to that of a living organism in which the presence of organs may play a major role in the development of common-sense intelligence (Lakoff & Johnson, 1999). Finally, **social** embodiment describes the fundamental relation between intelligent entities in the environment, and how these social relations may shape intelligence (Barsalou et al., 2003). In the context of robotics, other entities may be other (potentially heterogeneous) robots, humans, or even animals. Such interactions are extremely diverse and may be dictated by implicit or explicit social rules. All these notions manifest in robotics, but not necessarily in other domains of computer science and artificial intelligence. Thus, it is instructive to consider these notions to build better robotic systems.

Embodiment is not a problem in itself but an essential perspective for understanding the challenges of robotic applications. Being embodied in all the forms described above has many important implications. The physical and temporal views of embodiment, put together, imply a capacity to interact with the environment: the robot needs to perform actions to gather information and alter the environment. This introduces safety concerns, as actions in a physical environment may result in catastrophic outcomes. It also implies a highly dynamic range of environmental situations: in the real world, the people, objects, and furniture present in the room might change over time; a single type of object can be associated with many different forms and colours; and one task can be performed in different environmental settings (e.g., different ground textures or weather conditions) which may change during the robot's life. The robot's composite form entails the need to account for the inherent characteristics of its robotic parts: a particular sensor might require a specific behaviour to gather information properly, or any sensor or actuator may have flaws or momentarily malfunction. Finally, social embodiment implies the need to study the dynamics of interaction between multiple robots, as in MAS research, and between robots and humans (see Section 5.4).

It is important to note that, while physicality is an important aspect of embodiment, simulations are still a useful tool for studying embodiment. Simulated environments accurately replicate many challenges linked with embodiment: interaction with the environment, temporal embodiment, and social interactions. However, some other aspects are harder to emulate: accurate physical dynamics, variety and dynamicity of the environment, malfunctions, and human-robot

interactions. Thus, simulations are still relevant for addressing learning in robotics, but their shortages should not be overlooked and, if possible, addressed accordingly with real-world experiments.

Challenges in multi-agent learning and robotics are numerous and, when combined, often exacerbate each other, making multi-robot environments particularly challenging. Due to this great multiplicity of issues, it is common for some problems to be addressed separately. However, this leads to the domain of multi-agent learning research being highly fragmented, with many concurrent lines of work that are difficult to compare. To advance towards the learning of complex behaviours in multi-robot environments, multi-agent learning needs to integrate the inherent challenges of robotics and develop methods that efficiently tackle all issues of multi-agent learning. In the next two sections, we will review the main directions of multi-agent reinforcement learning research and then reflect on how they address the specific problems of robotics.

4 Methods in Multi-Agent (Deep) Reinforcement Learning

To tackle multi-agent environments, RL algorithms have been adapted and extended in various ways. As with single-agent RL, deep learning has enabled addressing more complex multi-agent environments. In this section, we present a survey of state-of-the-art MADRL algorithms. We focus on some important techniques employed in the past few years to produce efficient algorithms. Our objective is to present the different approaches and explain the functioning of state-of-the-art algorithms that are frequently found in the literature.

4.1 Multi-Agent Learning Paradigms

Designing a multi-agent learning algorithm requires first choosing how we consider one agent in relation to the MAS. This choice will dictate how information can be used for training the agent's policy and for executing it. In MASs, the gathering of information about the environment is always done locally by each agent, through the local observations o_t^i (see Figure 2). The processing of this information, however, can be done in various ways depending on our assumptions of how information can flow in the MAS. These assumptions will greatly impact the design of the resulting algorithms, as they dictate what information is available to them during training or execution. Here, we examine how, what, and when information can or should be centralised. By centralising information, we mean that the algorithm gathers the local observations together (i.e., it has access to the joint observation) and uses them during either the training phase to improve the learning update, the execution phase to improve local policies, or both. Note that this does not include communication, as defined in Section 2.3, that involves the agents actively choosing what information they should share with others. Centralising some information can greatly improve the efficiency of multi-agent learning algorithms. But, centralisation, if it is even possible, comes at several costs in terms of algorithmic conception and complexity. Thus, different MA(D)RL algorithms may prefer different levels of centralisation depending on environmental constraints and design decisions. They can be classified into three main categories that define how information is allowed to flow in the MAS during training and execution.

The **centralised training and execution** (CTE) paradigm allows information to be shared between agents at all times. This means that the agents' policies might be conditioned on information coming from other agents. Centralising information can take many different forms: sharing other agents' observations, internal hidden states, learnt parameters, generated value estimates, or policy outputs. This allows agents to generate better predictions and to have more information at hand during training to stabilise training. For example, a local policy conditioned on the joint observation, $a_{i,t} \sim \pi_i(.|\mathbf{o}_t)$, will have more information about the environment, allowing better prediction of the best action to choose. A more extreme version of this would have a single controller centralising all local observations and generating the joint action: $\mathbf{a}_t \sim \pi_{central}(.|\mathbf{o}_t)$. This would reduce the problem to a single-agent one, preventing the issues of multi-agent non-stationarity and credit assignment. But, these methods imply that the search spaces (for learnt estimates) grow exponentially with the number of agents, making such solutions poorly scalable. Additionally, in many environments, it might not even be possible to share information between agents. In realistic scenarios, agents are often independent and might not be connected to a central unit that can centralise and redistribute information freely.

At the opposite side of the spectrum, **decentralised training and execution** (DTE) considers that no information can ever be centralised. This is a far more realistic assumption as it does not rely on an algorithmic-level connection required to pass information between agents. It is also more computationally efficient to learn estimates in tighter observation and action spaces. So it allows better scaling to a large number of agents. However, because less information is at their disposal to train and compute estimation functions, these algorithms can learn less efficiently and suffer more from non-stationarity. Note, again, that this does not forbid having communication between agents. In fact, decentralised agents would largely benefit from learning an efficient way to discuss with partners to share local information and coordinate their actions (Cao et al., 2018).

To improve on DTE without losing the decentralised execution, a middle ground can be found with **centralised training and decentralised execution** (CTDE). In CTDE, information is allowed to be shared during training, but agents are kept totally independent during execution. Because training is usually done in a controlled environment, we can often make the reasonable assumption of being able to centralise some information during training. This allows a range of simplifications, with varying degrees of assumptions on how information can be centralised: from ensuring agents are trained on concurrent steps (Omidshafiei et al., 2017), to learning centralised value functions (see Section 4.3). In general, sharing some information makes RL training easier by reducing non-stationarity and compensating for partial observability. To ensure that execution is decentralised, the agents' policies are required to be conditioned only on local information, i.e., $a_i \sim pi_i(.|o_i)$. Because decentralised execution is such an important requirement and centralising information during training allows great improvements in RL training, the CTDE paradigm is the most preferred one today for MADRL algorithms.

4.2 Independent Learning

A seemingly simplistic approach to the multi-agent problem is to consider each agent totally independently of the rest of the MAS, ignoring other agents, as if they were parts of the environment. This is often referred to as Independent Learning (IL; Tan, 1993; Claus and Boutilier, 1998; Tampuu et al., 2017), where each agent can be trained independently with single-agent RL methods. But, a fully decentralised version of IL suffers badly from non-stationarity as the environment, composed of other learning agents, is continuously changing (Tan, 1993; Foerster et al., 2016a). Jiang and Lu, 2022 solve the problem of non-stationarity by learning from a surrogate transition probability that considers other agents will act optimally. But this requires strong assumptions on the environment, especially that it is deterministic. Thus, some assumptions from CTDE might be adopted to facilitate IL. For example, IL agents often share the parameters of their learnt policy or value functions (Tan, 1993; Foerster et al., 2016a; J. K. Gupta et al., 2017; Schroeder de Witt et al., 2020), allowing faster convergence. To maintain diverse behaviours, the policies are conditioned on a unique identification number corresponding to the agent. To enable the use of experience replay in a decentralised algorithm, Omidshafiei et al., 2017 introduced concurrent experience replay for training agents on the same environment steps during each update. This ensures that independent learners have a common training schedule, thus mitigating the effects of non-stationarity. While IL seems simplistic, it has actually been shown to be competitive with more centralised alternatives (Schroeder de Witt et al., 2020; Lyu et al., 2021; Jiang et al., 2024). This simplicity allows a great deal of freedom when designing training procedures. The self-sufficiency of agents makes it possible to modify the teams at will. For example, Jaderberg et al. (2019) trains a large population of independent agents, changing the team regularly so they learn to be robust to different partners and opponents.

4.3 Multi-Agent Actor-Critics

The CTDE framework requires only that the local policies are decentralised, that is, they must be conditioned on local information only (i.e., observations or history). Centralised information may be used, however, to improve the learning of these local policies. Therefore, a natural implementation of this paradigm is to use the actor-critic framework with a decentralised actor, i.e., a local policy conditioned on local observations, and a centralised critic, i.e., a value function conditioned on joint observations. The critic is used only during training to train the actor, using centralised information to improve the value estimates learnt during training. The **multi-agent deep deterministic policy gradient** (MADDPG; Lowe et al., 2017) introduced this approach by having one actor-critic for each agent, but allowing the critics to use the joint observation to compute the local action-value estimates (see Figure 3). By using information collected by all agents instead of only the local observation, the critic has more information on the current state of the environment to better estimate the action-value function. This also decreases the effects of non-stationarity: because the critic knows the state of other agents, it is less sensitive to changes in other agents' policies.

This centralised critic idea has been widely used for its intuitive advantages. It has been extended to include memory with a recurrent neural network used in both the decentralised policies and the centralised critic (R. E. Wang et al., 2020). Other actor-critic algorithms have been implemented this way, with *multi-agent proximal policy optimisation* (MAPPO; Yu et al., 2021b), *multi-agent twin-delayed DDPG* (MATD3; Ackermann et al., 2019) and *multi-agent soft actor-critic* (MASAC; Yu et al., 2021a). Iqbal and Sha (2019) added an attention mechanism in the centralised critic to better combine the centralised information. Finally, Foerster et al. (2018) used a centralised critic to improve credit assignment with their counterfactual multi-agent policy gradient (COMA). Having a centralised critic allows them to efficiently compute a marginal contribution for each local action, by computing the advantage of taking the action compared to all other possible actions: $A^i(\mathbf{h}, \mathbf{a}) = Q(\mathbf{h}, \mathbf{a}) - \sum_{a^{i\prime}} \pi^i(a^{i\prime}|h^i) Q(\mathbf{h}, (\mathbf{a}^{-i}, a^{i\prime}))$.

The nature of the centralised information used as input to the critic can vary. The joint observation is used often (Lowe et al., 2017; Iqbal & Sha, 2019). Memory-equipped agents can extend this with the joint history (Foerster et al., 2018; R. E. Wang et al., 2020; Yu et al., 2021b). But, it has also been proposed to use the global state of the environment

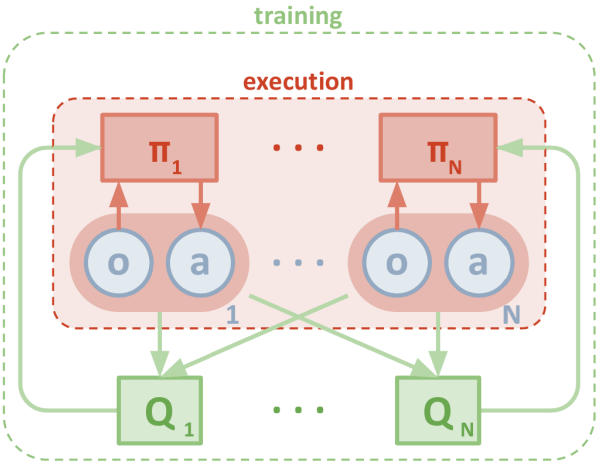


Figure 3: The MADDPG architecture (Lowe et al., 2017) illustrating the centralised critics training decentralised policies. Each agent has a local policy used during training and execution, and a centralised critic, conditioned on the joint observation, used only during training.

s_t, if it is defined and accessible (Lowe et al., 2017; Foerster et al., 2018). This is often described as preferable because the global state would be a more condensed and complete description of the environment compared to the joint observation. However, it has been shown that using the global state can result in a higher variance in training as a result of introducing bias (Lyu et al., 2023). Additionally, defining the global state is a problem on its own, which might not be straightforward in some environments. In fact, in rich and complex settings (e.g., the real world), it might be impossible to describe the state fully. Thus, choosing a particular state definition is a complex design choice that can have important impacts on training.

4.4 Value Factorisation

In Section 3.1.2, we presented the multi-agent credit assignment problem that arises when multiple agents share a common reward signal and need a way to measure the contribution of their behaviour towards the observed common outcome. In such settings with a common reward signal, learning the value of local action a^i, with respect to the global return G (as in Equation 2), can be problematic because G does not depend only on a^i, but also on the other agents' actions. In Section 3.1.2, we introduced methods that tackled this issue by computing a marginal contribution of each action. But, defining an effective marginal contribution can be tricky and computing it is usually expensive. Instead, another approach would be to learn the action-values of each action with respect to their real, unknown contribution, knowing that they are related in some way to the known common return. In other words, given the joint action-value $Q(\mathbf{h}, \mathbf{a})$ estimating the expected global return (see Equation 1) and local action-values $Q^i(h^i, a^i) \coloneqq \mathbb{E}_{\pi^i}[u^i \mid h^i, a^i]$, with u^i the *local utility* measuring the contribution of agent i in G, we need to find how the local values compose the joint value:

$$Q(\mathbf{h}, \mathbf{a}) = f(\{Q^i(h^i, a^i)\}_{1 \leq i \leq n}, \mathbf{h}). \tag{3}$$

The function f describes how each local value contributes to the joint value, depending on the joint history. Given the global return G, **value factorisation** (or "value decomposition") approaches learn the joint action-value and a way to decompose it into local action-values (i.e., function f). Value factorisation is a form of implicit credit assignment where we learn local value functions by learning how they compose the global value.

Learning this properly allows having local action-value functions that can be used for choosing greedy local actions for each agent. Because the objective is to maximise the global return, this requires that greedy local actions lead to optimal joint actions. This has been referred to as the **individual-global-max** (IGM; Rashid et al., 2018) property, requiring that choosing the greedy joint action with respect to the joint action-value corresponds to choosing greedy local actions with respect to each local action-value, i.e.:

$$\arg\max_{\mathbf{a}} Q(\mathbf{h}, \mathbf{a}) = \{\arg\max_{a^i} Q^i(h^i, a^i)\}_{1 \leq i \leq n}. \tag{4}$$

Respecting this property is essential to be able to use the learnt local action-values for local action selection. Thus, a multi-agent value-based learning algorithm that follows the IGM principle can fit into the CTDE paradigm, with decentralised local action-values used during execution, trained in a centralised manner with the help of a learnt joint action-value.

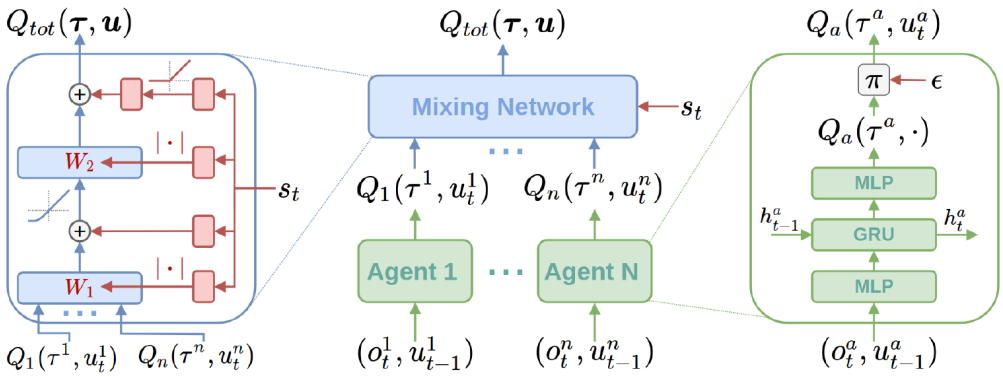


Figure 4: The architecture of QMIX (Rashid et al., 2018) illustrating the learnt monotonic value factorisation. The middle shows the overall architecture with local action-value functions, mixed in to compute the joint action-value. On the left is shown the mixing network that takes in the local action-values to compute the joint action-value, with the weights generated by the hypernetwork (in red) conditioned on the global state. On the right, the architecture for a local action-value function is described, with a recurrent network (GRU, for gated recurrent unit) used for memory of previous steps.

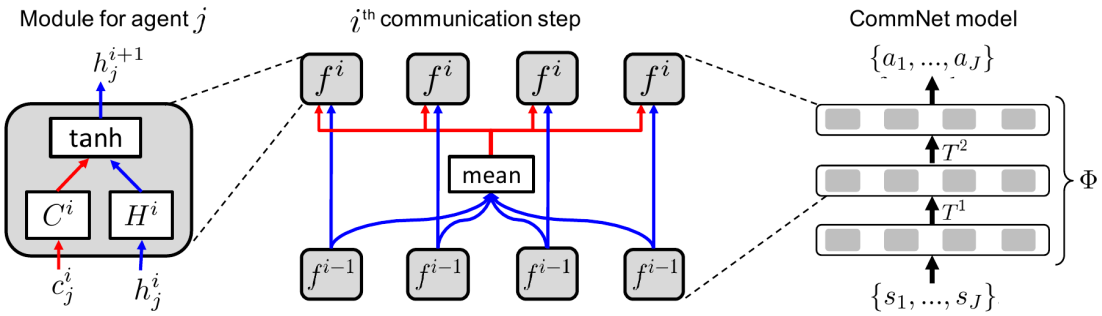
Sunehag et al., 2018 introduced this idea with their **value decomposition network** (VDN), making the simple assumption that local action-values should sum up to the joint action-value:

$$Q(\mathbf{h}, \mathbf{a}) \approx \sum_{i=1}^{n} Q^i(h^i, a^i). \tag{5}$$

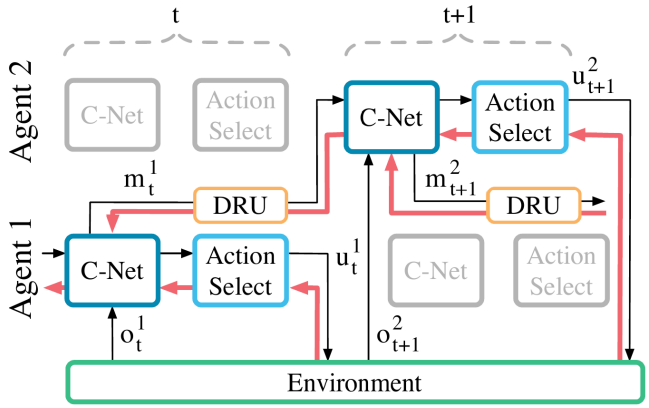
With this assumption, Q is differentiable with regard to each Q^i. Thus, VDN is able to learn local action-values by training optimising the objective of the deep Q-network (Mnih et al., 2015) computed on Q. This linear version of value factorisation has the advantage of simplicity and being computationally lightweight, which allows excellent scalability.

While this linear decomposition is intuitively logical and follows the IGM property, there is no guarantee that the true factorisation function f is a linear transformation of local utilities. Therefore, the formulation of VDN limits the factorisation operation and the learnt representations of joint and local action-values. To address this, **QMIX** (Rashid et al., 2018) introduces a separate "mixing" neural network that takes the local action-values and the global state as input, and outputs the joint action-value: $Q(s, \mathbf{a}) = f_{MIX}(\{Q^i(h^i, a^i)\}_{1 \leq i \leq n}, s)$ (see Figure 4). Note that, in their implementation, they consider that the global state s is available, but it can be replaced by the joint observation of history if needed. The mixing network learns a factorisation function that depends on the current state of the environment, allowing much richer factorisation capacities. To ensure that the IGM is respected, the mixing of local values must be monotonic: if a local value increases, the joint value must increase too, i.e., $\frac{\partial f_{MIX}}{\partial Q^i} \geq 0$. This monotonic constraint is ensured by having the weights of the mixing network be positive. But, this constraint must be applied only for the local action-values, not for the state. To allow this, QMIX employs a hypernetwork (Ha et al., 2017), which uses a separate MLP, conditioned on s to generate the weights of the MLP used for factorising the local action values. The absolute value of the generated weights is taken to ensure the monotonic constraint described above. Using a hypernetwork allows to depend on the state in a non-monotonic way and to learn more complex dependencies between the action values and the state (M. Zhou et al., 2020).

Many subsequent works have extended QMIX to improve its performance (Son et al., 2019; Rashid et al., 2020; Yang et al., 2020; M. Zhou et al., 2020; Iqbal et al., 2021; Peng et al., 2021; J. Wang, Ren, Liu, et al., 2021; Hong et al., 2022; H. Zhou et al., 2022; Sun et al., 2023; Xu et al., 2023). The main issue is the monotonic constraint that limits the potential of QMIX for modelling some factorisation functions, which might induce poor performance in some scenarios. Yang et al. (2020) reformulate the factorisation as a weighted sum that can be learnt with an attention mechanism. M. Zhou et al. (2020) and Peng et al. (2021) both extend QMIX to be used in an actor-critic algorithm. Having local policies for action-selection allows relaxing the monotonic constraint imposed in QMIX, required only because the local action-values were used for action-selection. Without this constraint, more accurate factorisation functions can be modelled. Additionally, using an actor-critic framework enables working with continuous actions (Peng et al., 2021) and learning stochastic policies (M. Zhou et al., 2020). QPLEX (J. Wang, Ren, Liu, et al., 2021) reformulates the problem by making the IGM property based on the advantage function instead of the action-value: given that $Q = V + A$ and that the action selection does not depend on V, the IGM constraint can be transferred onto the advantage function A, rewriting Equation 4 with the joint and individual advantages instead of action-values. This



(a) CommNet (Sukhbaatar et al., 2016)



(b) DIAL (Foerster et al., 2016b)

Figure 5: Two architectures of differentiable communication. (a) CommNet learns a centralised communication network that shares local information in multiple rounds of communication before choosing local actions. (b) DIAL learns decentralised communication with centralised training, by allowing gradients to flow between agents. Because messages result from differentiable operations of neural networks, communication can emerge from back-propagation of gradients from the RL objective.

allows easier learning of a value factorisation. Finally, Sun et al. (2023) propose a distributional extension of QMIX to better handle stochastic environments. While extensions provide several theoretical advantages, QMIX is still widely considered as state-of-the-art, and seen more often as a baseline. Also, it is interesting to note that the flexibility granted by QMIX for modelling more complex value factorisation might not be needed in many simpler tasks. In fact, VDN has been shown to outperform QMIX in some scenarios (Papoudakis, Christianos, Schäfer, & Albrecht, 2021; J. Wang, Ren, Han, et al., 2021).

4.5 Differentiable Emergent Communication

An important part of human interactions is communication. We often use our multi-modal communication abilities (e.g., speech, language, hand gestures, facial expressions) to share our knowledge, coordinate our actions, negotiate, or express goals and feelings. It is therefore natural to study communication in the context of multi-agent learning. We have seen, in Section 3.1, that the decentralised aspect of multi-agent settings gives rise to many issues that can be alleviated with efficient multi-agent communication. To achieve this, agents need to learn how to communicate. This may be summarised as learning *what to communicate* and *how to communicate it*. A popular approach for learning communication in a multi-agent setting is to let agents develop their own communication system in the process of learning to complete a given task. This gives rise to an emergent communication system that specifically fills the communication needs of the task.

Recently, a successful approach to emergent communication with MADRL has been to implement communication as a differentiable sub-step of the action-selection process (Foerster et al., 2016b; Sukhbaatar et al., 2016). By using only differentiable operations – i.e., neural networks – to generate and process the messages, the gradients of the RL objective can be back-propagated through the message-generating modules so they participate in maximising future returns. Importantly, having differentiable messages allows for gradients to be passed between agents: as a message generated by agent i will impact the choices made by agent j, agent i can be trained to generate messages that maximise

agent j's returns. This way, the communication mechanisms emerge from the task requirements. CommNet (Sukhbaatar et al., 2016) introduced a centralised approach of differentiable communication (see Figure 5a), with all agents sharing a communication network enabling information sharing and trained to maximise the joint return. DIAL (Foerster et al., 2016b) took this into the CTDE paradigm (see Figure 5b), with each agent's communication network (C-Net in the figure) trained from the other agent's learning objective. In these two approaches, neural networks generate differentiable messages comprised of a vector of continuous values, which may carry information to other agents. By learning to maximise the RL objective, the message-generating networks are trained to generate messages that allow other agents to select better actions.

This approach of differentiable communication has been extended in various ways for more targeted information sharing (Hoshen, 2017; Jiang & Lu, 2018; Das et al., 2019) or to limit bandwidth usage (A. Singh et al., 2019; Zhang et al., 2019; R. Wang et al., 2020; Han et al., 2023). While previously cited works use *continuous vectors* as messages, others have developed techniques to use *discrete symbols* for communicating (Cao et al., 2018; Lazaridou et al., 2018; Jaques et al., 2019; Kim et al., 2019; Rita et al., 2022). Discrete symbols are advantageous because they limit the bandwidth of transmitted messages. They also incite the emergence of certain qualities of natural languages that make human communications so efficient (Mordatch & Abbeel, 2018; Chaabouni et al., 2019).

While emergent communication allows efficient learning of information transmission with deep RL, it has important limitations. As with all deep learning approaches, it acts as a black box that lacks practical ways of interpreting (Lazaridou & Baroni, 2020). Defining metrics to measure the efficiency of emergent language and to understand how they are used by agents is challenging (Lowe et al., 2019). Because it emerges from task-oriented training in a closed group of agents, the resulting communication mechanisms will be strongly specialised on this training setting and will hardly generalise to other tasks and agents. For these reasons, methods are investigated to learn more interpretable and generalisable communication skills. One approach is the ground the communicated signals in external modalities, such as the observation space (Lowe et al., 2020; Lin et al., 2021; Karten et al., 2023) or language (Das et al., 2017; Havrylov & Titov, 2017; Lazaridou et al., 2020; H. Li et al., 2024), to ensure that messages carry meaning independent to the task. In Toquebiau et al. (2025), we proposed a different approach by training agents to learn a pre-defined language used for communication in embodied environments. We show that, as seen for humans, learning a language provides an efficient, generalisable communication strategy and helps for learning good representations of the world.

4.6 Agent Modelling

Previously presented approaches rely on learning policy and value functions to learn multi-agent behaviour. This model-free approach predominates in multi-agent learning because learning a model in a multi-agent setting is made extremely difficult by the fact that, from one agent's perspective, other agents contribute to the environment dynamics: the transition probability and reward function. One step towards solving this is to learn a model of other agents' policies, based on previous observations (Albrecht & Stone, 2018). In fictitious play (Brown, 1951; Robinson, 1951; Fudenberg & Levine, 1995; Hofbauer & Sandholm, 2002), each agent keeps track of action counts by other policies to compute potential action probabilities, then choosing an action accordingly. Recent deep RL techniques have been employed to improve fictitious play and allow its use in more complex, partially observable environments (Heinrich & Silver, 2016; Papoudakis, Christianos, & Albrecht, 2021; Strouse et al., 2021; Rahman et al., 2023; Jing et al., 2024). Bayesian learning goes further by tracking probabilities over possible policies for other agents, allowing to model uncertainty about their current reasoning (Jordan, 1991; Kalai & Lehrer, 1993; Bowling & Veloso, 2001; Foerster et al., 2019; Hu & Foerster, 2020). Such agent modelling approaches are promising for learning intricate multi-agent interactions, as they allow to adapt to the observed behaviour of other agents instead of trying to learn a policy able to effectively answer to any situation. Additionally, this approach has the intuitive advantage of emulating the way human beings approach their interactions with other intelligent entities, as described by the *theory of mind* literature (Apperly, 2011; Heyes & Frith, 2014; Aru et al., 2023).

5 Robotic Perspectives on MADRL Research: Open Challenges and Shortcomings

One of the key objectives of MADRL research is to facilitate the integration of robots into our daily lives. The real world is inherently multi-agent, as almost all conceivable situations involve interactions with other intelligent entities. Consequently, MADRL research aims to extend RL algorithms to be applied in complex multi-agent settings that more accurately reflect everyday scenarios. However, despite this objective, the current state of research has not yet come this far. The methods presented in the previous section are hardly applicable to robotic settings without significant modifications. This is partly because these methods are designed to be general multi-agent learning approaches, rather than being specifically designed for robotics. However, it may also result from some inherent limitations in their learning techniques or shortcomings of MADRL research.

In this section, we present four important challenges faced in MADRL research that must be addressed to enable the progress of MADRL algorithms in robotics. We define these challenges, examine how they are typically addressed, explore specific approaches to overcome them, and discuss potential improvements. While there may be other obstacles to overcome, we believe these challenges represent the main avenues for improving robotic control in complex multi-agent environments.

5.1 Benchmarking MADRL

Rigorous evaluation and comparison of different MADRL methods have been difficult to carry out due to several key challenges. Firstly, there is a large variety of learning environments and tasks, with little consensus on which setting should be used for studying which multi-agent problem. The most frequently found environments are the **Starcraft multi-agent challenge** (SMAC, see Figure 6a; Samvelyan et al., 2019) and the **multi-agent particle environment** (MPE, see Figure 6b; Lowe et al., 2017), but many others are also studied (see Figures 6c-f). Most environments have multiple tasks available for training and testing algorithms. But, it is often unclear what multi-agent learning problems are featured in a given task. Thus, different works choose different environments and tasks arbitrarily based on their preferences, available computing power, and the performance of their method. This complicates the comparison of different works that tackle different environments and tasks. Additionally, the value and rigour of these environments are seldom questioned, as shown by the recent revision of SMAC after it was found to be solvable by only observing the current time step (Ellis et al., 2023). Some other interesting environments are often proposed, for more efficient computation (Lechner et al., 2023; Michalski et al., 2023), human-agent teaming (Carroll et al., 2019), for allowing more agents (Lechner et al., 2023), more various tasks (Leroy et al., 2023), or more realistic settings (Kurach et al., 2020; Vinitsky et al., 2022); but there are seldom included in new studies and benchmarks.

Secondly, a thorough comparison with all existing methods is difficult. Learning multi-agent policies generally takes time and computing power. Among the available implementations, multiple versions of the same methods may have slight differences that are not always clearly stated. Different works might use different programming tools. While the programming language Python is widely adopted in Machine Learning, various Python libraries exist for implementing learning algorithms. No single library is universally preferred[2], leading to significant differences that prevent easy adaptation from one library to another.

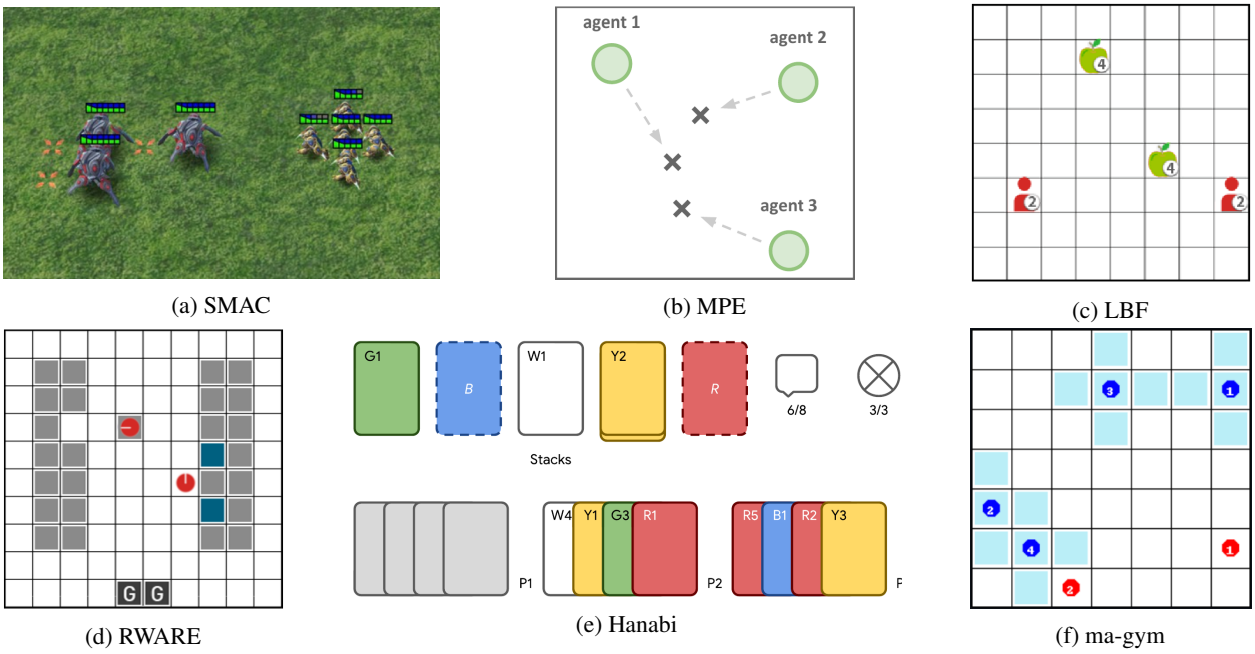
Lastly, the implementation of each method may differ from one work to another. All methods come with a very large set of hyperparameters, with some having a great impact on performance. Deep RL methods, which serve as the foundation for MADRL methods, can be implemented differently, with some implementation tricks having a major impact on performance. This variability makes comparison across different studies challenging.

For these reasons, assessing the progress of MADRL research is difficult. Performance reported in papers is hard to take at face value because of untold discrepancies hidden in the implementations and reported results (S. Singh et al., 2023). For example, Gorsane et al., 2022 show inconsistencies in the performance of QMIX reported in different papers. The consequence of this lack of standardised benchmark is concerning: it is unclear which methods are the best for any given purpose, and therefore what method should be used as a baseline in any given setting. To advance learning in multi-robot environments, it would be difficult to determine the most valuable MADRL algorithms to use in these environments.

Some benchmarks have been presented to try tackling this issue (Papoudakis, Christianos, Schäfer, & Albrecht, 2021; Yu et al., 2021a; Bettini et al., 2023; Ellis et al., 2023). They help clarify the field by providing a common ground for comparing important methods. However, there is limited variety in the environments used in these benchmarks, so the results might not hold in other tasks or more complex environments. It is also unclear what exact skill sets are required in each task, with only a rough measure of difficulty based on the returns obtained by all methods. This makes it difficult to discern the specific advantages of each method over others. Nevertheless, there are attempts to propose standardised evaluation protocols for new works (Gorsane et al., 2022; S. Singh et al., 2023), which is a promising avenue for building stronger and more progressive research in MADRL.

To move forward, better practices should be adopted. Proposed methods should all disclose hyperparameters and specific code-level optimisations. Evaluation protocols and metrics should be standardised across all new publications (Gorsane et al., 2022). Benchmarks should include more diverse environments, integrating the wide range of potential learning problems studied across multi-agent learning: continuous and discrete settings, various degrees of centralisation allowed during training and execution, communication between agents, and different degrees of environmental complexity with environments closer to robotics. There is no doubt that MADRL research would immensely benefit from improving its evaluation protocols as such, allowing less biased comparison between methods and deeper analysis of their abilities.

[2]Pytorch (https://pytorch.org/) is the most used in research, but some still use TensorFlow (https://www.tensorflow.org/), and a growing number of people prefer JAX for its computational efficiency (Bradbury et al., 2018).



(a) SMAC (b) MPE (c) LBF

(d) RWARE (e) Hanabi (f) ma-gym

Figure 6: The most used MADRL environments. (a) The *Starcraft multi-agent challenge* (SMAC; Samvelyan et al., 2019), with teams of units fighting opponents in various scenarios. (b) The *multi-agent particle environment* (MPE; Lowe et al., 2017), a two-dimensional continuous environment with various tasks studying cooperative navigation and communication. (c) The *level-based foraging* (LBF; Albrecht and Ramamoorthy, 2013) task, studying coordination in a grid world. (d) The *multi-robot warehouse* (RWARE; Christianos et al., 2020), a robotic task in a grid world. (e) The cooperative card game *Hanabi* (Bard et al., 2020) for studying complex team strategy learning and adaptation to teammates. (f) The *ma-gym* two-dimensional grid world environment, with various cooperative tasks.

This is essential for efficiently advancing the field and ensuring that proposed methods are robust and generalisable across tasks and environments.

5.2 Exploration

Exploration is arguably one of the most important problems in single-agent RL (Hao et al., 2024). It is particularly crucial when dealing with sparse rewards, where only a few interactions in the environment yield positive reinforcement signals. Such settings are often termed "hard-exploration" problems, requiring techniques allowing consistent discovery of the infrequent rewarding states. In multi-agent RL, the problem is exacerbated. Performance depends on the joint behaviour of all agents, requiring exploration of the space of joint policies to identify the best approaches. Exploration becomes a multi-agent problem, especially when coordination is necessary, as agents need to explore different ways to act in unison. Exploration is also a major subject in robotic environments. Partial observability complicates the issue, as one environment state may be observed from many different perspectives. Moreover, the safety concerns are both a constraint and an expectation for exploration. When exploration is conducted on robots, it should only involve safe states to ensure that robots do not injure themselves or others (Koller et al., 2018; Ding et al., 2021). At the same time, exploration is a way to find the optimal strategies that are safer for the robots. In this sense, exploration might be conducted in simulation to identify safe behaviours to execute on physical robots (García et al., 2015; Brunke et al., 2022).

Most single and multi-agent RL approaches treat exploration arbitrarily by infusing randomness into the behavioural policy during training. Q-learning-based approaches employ the epsilon-greedy strategy, policy-based approaches either add noise to actions, as seen in DDPG (Lillicrap et al., 2015), or maximise the entropy of the policy, as in PPO (Schulman et al., 2017). However, these methods are often insufficient for dealing with hard exploration problems (Ostrovski et al., 2017; Pathak et al., 2017; Burda et al., 2019). In multi-agent environments, random exploration often leads to the problem of relative overgeneralisation, where agents are attracted towards suboptimal Nash equilibria because the optimal strategy is too marginal to be found consistently through random exploration (Wiegand, 2003). In Toquebiau et al. (2024), we tackles this issue with joint intrinsic motivation to explicitly induce the exploration of coordinated behaviour.

5.3 Generalisation

Generalisation is a significant problem in machine learning and single-agent RL, concerning the robustness of learnt models to situations unseen during training. In RL, this may correspond to different initial conditions or new environmental settings. A good model is one that maintains its training performance in these new situations. In the multi-agent setting, the problem persists and even evolves with multiple agents, requiring to handle changes in the strategies of other agents.

In machine learning, good generalisation is typically achieved through extensive training on very large amounts of data. However, this becomes challenging when faced with embodiment issues (see Section 3.2.4). In RL, the training data is generated by the agents themselves. Thus, in multi-agent RL, acquiring a comprehensive understanding of the joint policy space is a challenging exploration problem, as discussed in the previous section. In multi-agent RL, agents usually train in "self-play", with a fixed team of agents learning by trying to solve the task together. However, this often leads to agents converging to an arbitrary convention on collective behaviour, which may not hold with new partners. This is a problem in robotic settings, where the environment is dynamic and robots are expected to efficiently and safely handle new robotic or human partners.

Generalising to new environmental situations can be facilitated by having diverse environmental settings, such as procedural maps, which allow training in many different scenarios and, hopefully, learning more general policies (Jaderberg et al., 2019; Cobbe et al., 2020). For generalising to different partners, one approach is population-based training, involving a large number of agents trained in dynamic teams to face various strategies during training (Jaderberg et al., 2019; S. Liu et al., 2019; Zhao et al., 2023). While this can be very effective (Jaderberg et al., 2019), it requires extensive training sessions to converge to general strategies. This may be impractical in high-dimensional and dynamic robotic environments, where training is expensive and accounting for all possible modifications of a real environment is impossible. Moreover, if humans are involved, it is impossible to train for all possible changes in human behaviour.

A promising approach is to learn to quickly adapt to any situation. In single-agent RL, *zero-shot generalisation* methods aim to adapt to unseen environmental settings without being retrained (Kirk et al., 2023; Haarnoja et al., 2024). In multi-agent RL, this concept extends to zero-shot, or *ad hoc*, teaming, where agents are evaluated with new partners (Stone et al., 2010). Agent modelling is promising for such settings, allowing to learn to model the "type" of policy observed in other agents (Strouse et al., 2021; Xie et al., 2021; Rahman et al., 2023; Yan et al., 2023), or even the exact agents faced (Barrett et al., 2017; Lanctot et al., 2023), allowing better reactions to the observed behaviour. If learned properly, zero-shot teaming can be a valuable tool for human-agent teaming. Training with humans is expensive, so it may be more efficient to learn to adapt quickly to any human partner (Shih et al., 2021; Strouse et al., 2021; Xie et al., 2021; Yan et al., 2023; Yu et al., 2023).

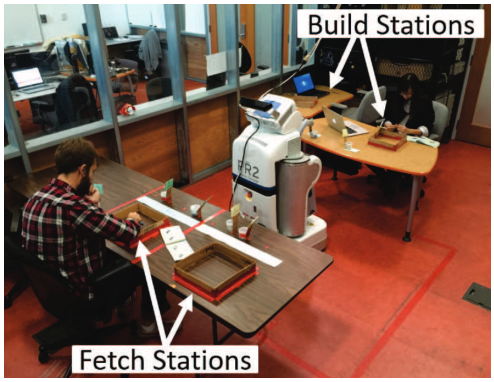
5.4 Interaction

Having robots in real environments implies the need for handling interactions with human beings. Figure 7 shows different examples of human-robot interaction (HRI), illustrating various levels of HRI with differing degrees of connection between robots and humans, ranging from simply living in a human-populated environment to deep social interactions between robots and humans. Across all these potential scenarios, we can identify three sub-problems of HRI. First, to enable interaction between humans and computer systems in general, we need to understand how humans behave. This means studying how they go about solving a task, to know how robots could help them and how they should not (Shih et al., 2021). This means understanding how they interact with each other, to understand what makes a successful human interaction and how robots could be a part of them (Tseng et al., 2016). This also means studying how humans react to robots when they interact with them, to understand the differences between human-human and human-robot interactions (Jung et al., 2020; Roesler et al., 2024). And, this means studying how humans communicate with each other, investigating different tools like natural language, body language, and a large variety of social cues (Feine et al., 2019). Analysing how humans behave in social interactions and when cooperating to fulfil a task can help design better robots and more effective learning approaches for HRI.

Second, HRI requires designing proper interfaces to enable smooth interactions. This involves physical interfaces, including the ability to sense, grasp, move, point and look at particular objects. These physical abilities are not needed only for interacting with human beings, but the HRI component might influence the design of these skills. Social interfaces are also required, with the help of human-like features such as voice, eyes, articulated faces, and gestures (Złotowski et al., 2014). Being able to communicate with humans is also an important requirement to allow information sharing, teaching, strategy evaluation and correction (Crandall et al., 2018; Mikolov et al., 2018), and, more generally to bond more easily with artificial agents (C.-C. Liu et al., 2022).

Finally, HRI requires learning to behave around humans and cooperate with them. This is where MADRL research becomes relevant, as it enables learning in complex environments with multiple intelligent entities interacting. However,

(a)



(b)

(c)

Figure 7: Examples of human-robot interaction. (a) Navigation of a robot in a human-populated environment (Chen et al., 2017). (b) A robot assisting humans in a building task by fetching objects (Gombolay et al., 2017). (c) A robot helping a child with autism spectrum disorder to learn social interaction (Scassellati et al., 2018).

the main MADRL algorithms, discussed in Section 4, are not specifically intended for interacting with humans. They need to be extended using various techniques to efficiently address the problems linked with embodiment (see Section 3.2.4). The generalisation problem, described previously, must be tackled to enable robots to interact with any given human partner. And, to enable the use of MADRL algorithms on real robots, we need efficient approaches for bridging the reality gap (see Section 3.2.3).

6 Conclusion

In this article, we explored the domain of MADRL research from the perspective of robotics. One of the objectives of MADRL research is to learn behavioural policies for controlling robots in the real world. Thus, it is important to reflect on the current state of progress in this domain, analyse the relevance of recent studies for robotics, and see how the challenges of robotic environments are addressed. After formally defining the tools of MADRL, we have presented the main challenges faced in multi-agent learning and robotics. Some of these challenges are specific to one of the two areas, but many overlap on many aspects. This analysis shows that connecting these two domains means dealing with many different forms of complexity, in environmental settings, interactions, and unpredictable situations.

In Section 4, we presented a survey of the main avenues of MADRL research, describing how state-of-the-art methods learn policies in multi-agent settings. While some techniques have been shown to tackle increasingly complex tasks and environments, the main approaches are still far from being applicable in a robotic scenario without significant adaptation. This can be attributed to the fact that MADRL research is focused on the optimisation problem of multi-agent learning, i.e., finding an efficient multi-agent strategy. Doing so, it often overlooks some problems faced in realistic scenarios and thus fails to progress towards more efficient learning algorithms.

In Section 5, we introduced multiple challenges for MADRL research that should be addressed for improving the field, especially for moving towards robotic applications. The benchmarking issue faced in the domain is especially important to solve rapidly to ensure a more reliable research field and more steady progress. Next, the problems of exploration, generalisation, and interaction, are all key to improving the efficiency and applicability of new MADRL algorithms. We believe these challenges represent the main directions for advancing the control of robots in complex multi-agent environments, and that MADRL algorithms would benefit from investigating them further.

References

Abbeel, P., Coates, A., Quigley, M., & Ng, A. (2006). *An application of reinforcement learning to aerobatic helicopter flight* (B. Schölkopf, J. Platt, & T. Hoffman, Eds.).

Ackermann, J., Gabler, V., Osa, T., & Sugiyama, M. (2019). *Reducing overestimation bias in multi-agent domains using double centralized critics* [arxiv:1910.01465].

Albrecht, S. V., & Ramamoorthy, S. (2013). *A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems*. *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, 1155–1156.

Albrecht, S. V., & Stone, P. (2018). *Autonomous agents modelling other agents: A comprehensive survey and open problems*. *Artificial Intelligence*, *258*, 66–95.

Amato, C., Konidaris, G., Kaelbling, L. P., & How, J. P. (2019). *Modeling and planning with macro-actions in decentralized pomdps*. *Journal of Artificial Intelligence Research*, *64*, 817–859.

Andrychowicz, O. M., Baker, B., Chociej, M., Józefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., & Zaremba, W. (2020). *Learning dexterous in-hand manipulation*. *The International Journal of Robotics Research*, *39*(1), 3–20.

Apperly, I. (2011). *Mindreaders: The cognitive basis of "theory of mind"*. Psychology Press.

Aru, J., Labash, A., Corcoll, O., & Vicente, R. (2023). *Mind the gap: Challenges of deep learning approaches to theory of mind*. *Artificial Intelligence Review*, *56*(9), 9141–9156.

Austin, J. L. (1975). *How to do things with words* (J. O. Urmson & M. Sbisà, Eds.). Harvard university press.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., Dunning, I., Mourad, S., Larochelle, H., Bellemare, M. G., & Bowling, M. (2020). *The hanabi challenge: A new frontier for ai research*. *Artificial Intelligence*, *280*, 103216.

Barrett, S., Rosenfeld, A., Kraus, S., & Stone, P. (2017). *Making friends on the fly: Cooperating with new teammates*. *Artificial Intelligence*, *242*, 132–171.

Barsalou, L. W., Niedenthal, P. M., Barbey, A. K., & Ruppert, J. A. (2003). *Social embodiment*. In *Psychology of learning and motivation volume 43* (pp. 43–92). Elsevier.

Ben-Ari, M. (2006). *Principles of concurrent and distributed programming (2nd edition)*. Addison-Wesley Longman Publishing Co., Inc.

Bettini, M., Prorok, A., & Moens, V. (2023). *Benchmarl: Benchmarking multi-agent reinforcement learning* [arxiv:2312.01472].

Böhmer, W., Kurin, V., & Whiteson, S. (2020). *Deep coordination graphs*. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 980–991, Vol. 119). PMLR.

Bousquet, F., & Le Page, C. (2004). *Multi-agent simulations and ecosystem management: A review*. *Ecological Modelling*, *176*(3–4), 313–332.

Bowling, M., & Veloso, M. (2001). *Rational and convergent learning in stochastic games*. *International joint conference on artificial intelligence*, *17*(1), 1021–1026.

Bowling, M., & Veloso, M. (2002). *Multiagent learning using a variable learning rate*. *Artificial Intelligence*, *136*(2), 215–250.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.3.13).

Brighton, H., Smith, K., & Kirby, S. (2005). *Language as an evolutionary system*. *Physics of Life Reviews*, *2*(3), 177–226.

Brown, G. W. (1951). *Iterative solution of games by fictitious play*. *Proceedingsof the Conference on Activity Analysis of Production and Allocation, Cowles CommissionMonograph 13*, 374–376.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., & Schoellig, A. P. (2022). *Safe learning in robotics: From learning-based control to safe reinforcement learning*. *Annual Review of Control, Robotics, and Autonomous Systems*, *5*(1), 411–444.

Burda, Y., Edwards, H., Storkey, A., & Klimov, O. (2019). *Exploration by random network distillation* [arxiv:1810.12894]. *7th International Conference on Learning Representations*.

Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., & Clark, S. (2018). *Emergent communication through negotiation*. *6th International Conference on Learning Representations*.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., & Dragan, A. (2019). *On the utility of learning about humans for human-ai coordination*. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

Chaabouni, R., Kharitonov, E., Dupoux, E., & Baroni, M. (2019). *Anti-efficient encoding in emergent communication*. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N., & Fox, D. (2019). *Closing the sim-to-real loop: Adapting simulation randomization with real world experience*. *2019 International Conference on Robotics and Automation (ICRA)*.

Chen, Y. F., Liu, M., Everett, M., & How, J. P. (2017). *Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning*. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 285–292.

Christianos, F., Schäfer, L., & Albrecht, S. (2020). *Shared experience actor-critic for multi-agent reinforcement learning*. *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 10707–10717.

Claus, C., & Boutilier, C. (1998). *The dynamics of reinforcement learning in cooperative multiagent systems*. *AAAI/IAAI*, *1998*(746-752), 2.

Cobbe, K., Hesse, C., Hilton, J., & Schulman, J. (2020). *Leveraging procedural generation to benchmark reinforcement learning*. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 2048–2056, Vol. 119). PMLR.

Conitzer, V., & Sandholm, T. (2007). *Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents*. *Machine Learning*, *67*(1-2), 23–43.

Crandall, J. W., Oudah, M., Tennom, Ishowo-Oloko, F., Abdallah, S., Bonnefon, J.-F., Cebrian, M., Shariff, A., Goodrich, M. A., & Rahwan, I. (2018). *Cooperating with machines*. *Nature Communications*, *9*(1).

D'Ambrosio, D. B., Abeyruwan, S., Graesser, L., Iscen, A., Amor, H. B., Bewley, A., Reed, B. J., Reymann, K., Takayama, L., Tassa, Y., Choromanski, K., Coumans, E., Jain, D., Jaitly, N., Jaques, N., Kataoka, S., Kuang, Y., Lazic, N., Mahjourian, R., … Sanketi, P. R. (2024). *Achieving human level competitive robot table tennis* [arxiv:2408.03906].

Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., & Pineau, J. (2019). *TarMAC: Targeted multi-agent communication*. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 1538–1546, Vol. 97). PMLR.

Das, A., Kottur, S., Moura, J. M. F., Lee, S., & Batra, D. (2017). *Learning cooperative visual dialog agents with deep reinforcement learning*. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Ding, D., Wei, X., Yang, Z., Wang, Z., & Jovanovic, M. (2021). *Provably efficient safe exploration via primal-dual policy optimization*. In A. Banerjee & K. Fukumizu (Eds.), *Proceedings of the 24th international conference on artificial intelligence and statistics* (pp. 3304–3312, Vol. 130). PMLR.

Doran, J., & Palmer, M. (1995). *The eos project: Integrating two models of palaeolithic social change*. *Artificial Societies: The ComputerSimulation of Social Life*, 103–125.

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., … Florence, P. (2023). *Palm-e: An embodied multimodal language model*. In PMLR (Ed.), *Proceedings of the 40th international conference on machine learning* (Vol. 202).

Ellis, B., Cook, J., Moalla, S., Samvelyan, M., Sun, M., Mahajan, A., Foerster, J., & Whiteson, S. (2023). *Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning*. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (pp. 37567–37593, Vol. 36). Curran Associates, Inc.

Farrell, J., & Rabin, M. (1996). *Cheap talk*. *Journal of Economic Perspectives*, *10*(3), 103–118.

Fatima, S. S., Wooldridge, M., & Jennings, N. R. (2008). *A linear approximation method for the shapley value*. *Artificial Intelligence*, *172*(14), 1673–1699.

Feine, J., Gnewuch, U., Morana, S., & Maedche, A. (2019). *A taxonomy of social cues for conversational agents*. *International Journal of Human-Computer Studies*, *132*, 138–161.

Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016a). *Learning to communicate to solve riddles with deep distributed recurrent q-networks* [arxiv:1602.02672].

Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016b). *Learning to communicate with deep multi-agent reinforcement learning*. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2145–2153.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018). *Counterfactual multi-agent policy gradients*. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*.

Foerster, J. N., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H. S., Kohli, P., & Whiteson, S. (2017). *Stabilising experience replay for deep multi-agent reinforcement learning*. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (pp. 1146–1155, Vol. 70). PMLR.

Foerster, J. N., Song, F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., Botvinick, M., & Bowling, M. (2019). *Bayesian action decoder for deep multi-agent reinforcement learning*. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 1942–1951, Vol. 97). PMLR.

Fudenberg, D., & Levine, D. K. (1995). *Consistency and cautious fictitious play*. *Journal of Economic Dynamics and Control*, *19*(5–7), 1065–1089.

Galke, L., Ram, Y., & Raviv, L. (2022). *Emergent communication for understanding human language evolution: What's missing?* *Emergent Communication Workshop at ICLR 2022*.

García, J., Fern, & o Fernández. (2015). *A comprehensive survey on safe reinforcement learning*. *Journal of Machine Learning Research*, *16*(42), 1437–1480.

Gombolay, M., Bair, A., Huang, C., & Shah, J. (2017). *Computational design of mixed-initiative human–robot teaming that considers human factors: Situational awareness, workload, and workflow preferences*. *The International Journal of Robotics Research*, *36*(5–7), 597–617.

Gorsane, R., Mahjoub, O., de Kock, R. J., Dubb, R., Singh, S., & Pretorius, A. (2022). *Towards a standardised performance evaluation protocol for cooperative marl*. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 5510–5521, Vol. 35). Curran Associates, Inc.

Guestrin, C., Lagoudakis, M., & Parr, R. (2002). *Coordinated reinforcement learning*. *Proceedings of 19th International Conference on Machine Learning*, *2*, 227–234.

Gupta, A., Devin, C., Liu, Y., Abbeel, P., & Levine, S. (2017). *Learning invariant feature spaces to transfer skills with reinforcement learning*. *International Conference on Learning Representations*.

Gupta, J. K., Egorov, M., & Kochenderfer, M. (2017). *Cooperative multi-agent control using deep reinforcement learning*. In *Autonomous agents and multiagent systems* (pp. 66–83). Springer International Publishing.

Ha, D., Dai, A., & Le, Q. V. (2017). *Hypernetworks*.

Haarnoja, T., Moran, B., Lever, G., Huang, S. H., Tirumala, D., Humplik, J., Wulfmeier, M., Tunyasuvunakool, S., Siegel, N. Y., Hafner, R., Bloesch, M., Hartikainen, K., Byravan, A., Hasenclever, L., Tassa, Y., Sadeghi, F., Batchelor, N., Casarini, F., Saliceti, S., … Heess, N. (2024). *Learning agile soccer skills for a bipedal robot with deep reinforcement learning*. *Science Robotics*, *9*(89), eadi8022.

Hamann, H. (2018). *Swarm robotics: A formal approach*. Springer International Publishing.

Hamill, L., & Gilbert, N. (2015). *Agent-based modelling in economics*. John Wiley & Sons.

Han, S., Dastani, M., & Wang, S. (2023). *Model-based sparse communication in multi-agent reinforcement learning*. *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 439–447.

Hao, J., Yang, T., Tang, H., Bai, C., Liu, J., Meng, Z., Liu, P., & Wang, Z. (2024). *Exploration in deep reinforcement learning: From single-agent to multiagent domain*. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(7), 8762–8782.

Harsanyi, J. C., & Selten, R. (1988). *A General Theory of Equilibrium Selection in Games* (Vol. 1). The MIT Press.

Hausknecht, M., & Stone, P. (2015). *Deep recurrent q-learning for partially observable mdps*. *Sequential Decision Making for Intelligent Agents Papers from the AAAI 2015 Fall Symposium*.

Havrylov, S., & Titov, I. (2017). *Emergence of language with multi-agent games: Learning to communicate with sequences of symbols*. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2146–2156.

Heinrich, J., & Silver, D. (2016). *Deep reinforcement learning from self-play in imperfect-information games* [arxiv:1603.01121].

Heyes, C. M., & Frith, C. D. (2014). *The cultural evolution of mind reading*. *Science*, *344*(6190).

Hofbauer, J., & Sandholm, W. H. (2002). *On the global convergence of stochastic fictitious play*. *Econometrica*, *70*(6), 2265–2294.

Hong, Y., Jin, Y., & Tang, Y. (2022). *Rethinking individual global max in cooperative multi-agent reinforcement learning*. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 32438–32449, Vol. 35). Curran Associates, Inc.

Hoshen, Y. (2017). *Vain: Attentional multi-agent predictive modeling*. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2698–2708.

Hu, H., & Foerster, J. N. (2020). *Simplified action decoder for deep multi-agent reinforcement learning*. *International Conference on Learning Representations*.

Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., & Levine, S. (2021). *How to train your robot with deep reinforcement learning: Lessons we have learned*. *The International Journal of Robotics Research*, *40*(4-5).

Iqbal, S., De Witt, C. A. S., Peng, B., Boehmer, W., Whiteson, S., & Sha, F. (2021). *Randomized entity-wise factorization for multi-agent reinforcement learning*. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (pp. 4596–4606, Vol. 139). PMLR.

Iqbal, S., & Sha, F. (2019, September). *Actor-attention-critic for multi-agent reinforcement learning*. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 2961–2970, Vol. 97). PMLR.

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., Sonnerat, N., Green, T., Deason, L., Leibo, J. Z., Silver, D., Hassabis, D., Kavukcuoglu, K., & Graepel, T. (2019). *Human-level performance in first-person multiplayer games with population-based deep reinforcement learning*. *Science*, *364*(6443), 859–865.

Jakobi, N., Husbands, P., & Harvey, I. (1995). *Noise and the reality gap: The use of simulation in evolutionary robotics*. In *Advances in artificial life* (pp. 704–720). Springer Berlin Heidelberg.

James, S., Wohlhart, P., Kalakrishnan, M., Kalashnikov, D., Irpan, A., Ibarz, J., Levine, S., Hadsell, R., & Bousmalis, K. (2019). *Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P. A., Strouse, D., Leibo, J. Z., & de Freitas, N. (2019). *Social influence as intrinsic motivation for multi-agent deep reinforcement learning*. *Proceedings of the 36th International Conference on Machine Learning*, *97*, 3040–3049.

Jiang, J., & Lu, Z. (2018). *Learning attentional communication for multi-agent cooperation*. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 31). Curran Associates, Inc.

Jiang, J., & Lu, Z. (2022). *I2q: A fully decentralized q-learning algorithm*. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 20469–20481, Vol. 35). Curran Associates, Inc.

Jiang, J., Su, K., & Lu, Z. (2024). *Fully decentralized cooperative multi-agent reinforcement learning: A survey* [arxiv:2401.04934].

Jing, Y., Li, K., Liu, B., Zang, Y., Fu, H., FU, Q., Xing, J., & Cheng, J. (2024). *Towards offline opponent modeling with in-context learning*. *The Twelfth International Conference on Learning Representations*.

Jordan, J. (1991). *Bayesian learning in normal form games*. *Games and Economic Behavior*, *3*(1), 60–81.

Ju, H., Juan, R., Gomez, R., Nakamura, K., & Li, G. (2022). *Transferring policy of deep reinforcement learning from simulation to reality for robotics*. *Nature Machine Intelligence*, *4*(12), 1077–1087.

Jung, M. F., Difranzo, D., Shen, S., Stoll, B., Claure, H., & Lawrence, A. (2020). *Robot-assisted tower construction—a method to study the impact of a robot's allocation behavior on interpersonal dynamics and collaboration in groups*. *ACM Transactions on Human-Robot Interaction*, *10*(1), 1–23.

Kalai, E., & Lehrer, E. (1993). *Rational learning leads to nash equilibrium*. *Econometrica*, *61*(5), 1019.

Karten, S., Kailas, S., & Sycara, K. (2023). *Emergent compositional concept communication through mutual information in multi-agent teams*. *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2391–2393.

Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K., & Yi, Y. (2019). *Learning to schedule communication in multi-agent reinforcement learning*. *International Conference on Learning Representations*.

Kirk, R., Zhang, A., Grefenstette, E., & Rocktäschel, T. (2023). *A survey of zero-shot generalisation in deep reinforcement learning*. *Journal of Artificial Intelligence Research*, *76*, 201–264.

Kiverstein, J. (2012). *The meaning of embodiment*. *Topics in Cognitive Science*, *4*(4), 740–758.

Koller, T., Berkenkamp, F., Turchetta, M., & Krause, A. (2018). *Learning-based model predictive control for safe exploration*. *2018 IEEE Conference on Decision and Control (CDC)*.

Kurach, K., Raichuk, A., Stańczyk, P., Zając, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., & Gelly, S. (2020). *Google research football: A novel reinforcement learning environment*. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 4501–4510.

Lakoff, G., & Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*. New York: Basic Books.

Lanctot, M., Schultz, J., Burch, N., Smith, M. O., Hennes, D., Anthony, T., & Perolat, J. (2023). *Population-based evaluation in repeated rock-paper-scissors as a benchmark for multiagent reinforcement learning*. *Transactions on Machine Learning Research*.

Lazaridou, A., & Baroni, M. (2020). *Emergent multi-agent communication in the deep learning era* [arxiv:2006.02419].

Lazaridou, A., Hermann, K. M., Tuyls, K., & Clark, S. (2018). *Emergence of linguistic communication from referential games with symbolic and pixel input*. *International Conference on Learning Representations (ICLR)*.

Lazaridou, A., Potapenko, A., & Tieleman, O. (2020). *Multi-agent communication meets natural language: Synergies between functional and structural language learning*. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7663–7674.

Lechner, M., yin lianhao, l., Seyde, T., Wang, T.-H. J., Xiao, W., Hasani, R., Rountree, J., & Rus, D. (2023). *Gigastep - one billion steps per second multi-agent reinforcement learning*. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (pp. 155–170, Vol. 36). Curran Associates, Inc.

Lee, A. X., Nagabandi, A., Abbeel, P., & Levine, S. (2019). *Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model*. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 741–752, Vol. 33). Curran Associates, Inc.

Leroy, P., Morato, P. G., Pisane, J., Kolios, A., & Ernst, D. (2023). *Imp-marl: A suite of environments for large-scale infrastructure management planning via marl*. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (pp. 53522–53551, Vol. 36). Curran Associates, Inc.

Leyton-Brown, K., & Shoham, Y. (2008). *Essentials of game theory: A concise, multidisciplinary introduction*. Springer International Publishing.

Li, H., Mahjoub, H. N., Chalaki, B., Tadiparthi, V., Lee, K., Pari, E. M., Lewis, C. M., & Sycara, K. P. (2024). *Language grounded multi-agent reinforcement learning with human-interpretable communication*. *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Li, S., Gupta, J. K., Morales, P., Allen, R., & Kochenderfer, M. J. (2021). *Deep implicit coordination graphs for multi-agent reinforcement learning*. *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 764–772.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). *Continuous control with deep reinforcement learning*.

Lin, T., Huh, M., Stauffer, C., Lim, S.-N., & Isola, P. (2021). *Learning to ground multi-agent communication with autoencoders*. *Advances in Neural Information Processing Systems*.

Liu, C.-C., Liao, M.-G., Chang, C.-H., & Lin, H.-M. (2022). *An analysis of children' interaction with an ai chatbot and its impact on their interest in reading*. *Computers & Education*, *189*, 104576.

Liu, S., Lever, G., Merel, J., Tunyasuvunakool, S., Heess, N., & Graepel, T. (2019). *Emergent coordination through competition*. *International Conference on Learning Representations*.

Lowe, R., Foerster, J., Boureau, Y.-L., Pineau, J., & Dauphin, Y. (2019). *On the pitfalls of measuring emergent communication*. *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 693–701.

Lowe, R., Gupta, A., Foerster, J., Kiela, D., & Pineau, J. (2020). *On the interaction between supervision and self-play in emergent communication*. *International Conference on Learning Representations*.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). *Multi-agent actor-critic for mixed cooperative-competitive environments*. *Advances in Neural Information Processing Systems*, *30*, 1–12.

Lyu, X., Baisero, A., Xiao, Y., Daley, B., & Amato, C. (2023). *On centralized critics in multi-agent reinforcement learning*. *Journal of Artificial Intelligence Research*, *77*, 295–354.

Lyu, X., Xiao, Y., Daley, B., & Amato, C. (2021). *Contrasting centralized and decentralized critics in multi-agent reinforcement learning*. *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 844–852.

Michalak, T. P., Aadithya, K. V., Szczepanski, P. L., Ravindran, B., & Jennings, N. R. (2014). *Efficient computation of the shapley value for game-theoretic network centrality*. *Journal Of Artificial Intelligence Research, Volume 46, pages 607-650, 2013*.

Michalski, A., Christianos, F., & Albrecht, S. V. (2023). *Smaclite: A lightweight environment for multi-agent reinforcement learning* [arxiv:2305.05566].

Mikolov, T., Joulin, A., & Baroni, M. (2018). *A roadmap towards machine intelligence*. In *Lecture notes in computer science* (pp. 29–61). Springer International Publishing.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). *Human-level control through deep reinforcement learning*. *Nature 518*, *518*(7540), 529–533.

Mordatch, I., & Abbeel, P. (2018). *Emergence of grounded compositional language in multi-agent populations*. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*, 1495–1502.

Nachum, O., Ahn, M., Ponte, H., Gu, S. (, & Kumar, V. (2020). *Multi-agent manipulation via locomotion using hierarchical sim2real* (L. P. Kaelbling, D. Kragic, & K. Sugiura, Eds.).

Nash, J. F. (1950). *Equilibrium points in n-person games*. *Proceedings of the National Academy of Sciences*, *36*(1), 48–49.

Nowé, A., Vrancx, P., & De Hauwere, Y.-M. (2012). *Game theory and multi-agent reinforcement learning*. In M. Wiering & M. van Otterlo (Eds.), *Reinforcement learning: State-of-the-art* (pp. 441–470). Springer Berlin Heidelberg.

Oliehoek, F. A., & Amato, C. (2016). *A concise introduction to decentralized pomdps*. Springer.
Omidshafiei, S., Pazis, J., Amato, C., How, J. P., & Vian, J. (2017). *Deep decentralized multi-task multi-agent reinforcement learning under partial observability*. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (pp. 2681–2690, Vol. 70). PMLR.
OpenAI, Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., d. O. Pinto, H. P., Raiman, J., Salimans, T., … Zhang, S. (2019). *Dota 2 with large scale deep reinforcement learning* [arXiv:1912.06680].
Orr, J., & Dutta, A. (2023). *Multi-agent deep reinforcement learning for multi-robot applications: A survey*. *Sensors*, *23*(7), 3625.
Ostrovski, G., Bellemare, M. G., van den Oord, A., & Munos, R. (2017). *Count-based exploration with neural density models*. *Proceedings of the 34th International Conference on Machine Learning*, *70*, 2721–2730.
Owen, G. (2013). *Game theory*. Emerald Group Publishing.
Papadimitriou, C. H., & Tsitsiklis, J. N. (1987). *The complexity of markov decision processes*. *Mathematics of Operations Research*, *12*(3), 441–450.
Papoudakis, G., Christianos, F., & Albrecht, S. (2021). *Agent modelling under partial observability for deep reinforcement learning*. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (pp. 19210–19222, Vol. 34). Curran Associates, Inc.
Papoudakis, G., Christianos, F., Schäfer, L., & Albrecht, S. V. (2021). *Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks*. *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
Parker, L. E., Rus, D., & Sukhatme, G. S. (2016). *Multiple mobile robot systems*. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (pp. 1335–1384). Springer International Publishing.
Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). *Curiosity-driven exploration by self-supervised prediction*. *Proceedings of the 34th International Conference on Machine Learning*, *70*, 2778–2787.
Peng, B., Rashid, T., Schroeder de Witt, C., Kamienny, P.-A., Torr, P., Boehmer, W., & Whiteson, S. (2021). *Facmac: Factored multi-agent centralised policy gradients*. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (pp. 12208–12221, Vol. 34). Curran Associates, Inc.
Petersen, K., Nagpal, R., & Werfel, J. (2012). *Termes: An autonomous robotic system for three-dimensional collective construction*. In *Robotics* (pp. 257–264). The MIT Press.
Pfeifer, R., & Bongard, J. (2006). *How the body shapes the way we think: A new view of intelligence*. MIT Press.
Pierson, H. A., & Gashler, M. S. (2017). *Deep learning in robotics: A review of recent research*. *Advanced Robotics*, *31*(16), 821–835.
Pinto, L., & Gupta, A. (2016). *Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours*. *2016 IEEE International Conference on Robotics and Automation (ICRA)*.
Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). *Learning transferable visual models from natural language supervision*. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (pp. 8748–8763, Vol. 139). PMLR.
Rahman, A., Carlucho, I., Höpner, N., & Albrecht, S. V. (2023). *A general learning framework for open ad hoc teamwork using graph-based policy learning*. *Journal of Machine Learning Research*, *24*(298), 1–74.
Rashid, T., Farquhar, G., Peng, B., & Whiteson, S. (2020). *Weighted QMIX: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning*. *Advances in Neural Information Processing Systems 33*, 10199–10210.
Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., & Whiteson, S. (2018). *QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning*. *Proceedings of the 35th International Conference on Machine Learning*, *80*, 4295–4304.
Rita, M., Tallec, C., Michel, P., Grill, J.-B., Pietquin, O., Dupoux, E., & Strub, F. (2022). *Emergent communication: Generalization and overfitting in lewis games*. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 1389–1404, Vol. 35). Curran Associates, Inc.
Rizk, Y., Awad, M., & Tunstel, E. W. (2019). *Cooperative heterogeneous multi-robot systems: A survey*. *ACM Computing Surveys (CSUR)*, *52*(2), 1–31.
Robinson, J. (1951). *An iterative method of solving a game*. *The Annals of Mathematics*, *54*(2), 296.
Roesler, E., Vollmann, M., Manzey, D., & Onnasch, L. (2024). *The dynamics of human–robot trust attitude and behavior — exploring the effects of anthropomorphism and type of failure*. *Computers in Human Behavior*, *150*, 108008.
Rosenschein, J. S., & Zlotkin, G. (1994). *Rules of encounter*. The MIT Press.

Rusu, A. A., Večerík, M., Rothörl, T., Heess, N., Pascanu, R., & Hadsell, R. (2017). *Sim-to-real robot learning from pixels with progressive nets*. In S. Levine, V. Vanhoucke, & K. Goldberg (Eds.), *Proceedings of the 1st annual conference on robot learning* (pp. 262–270, Vol. 78). PMLR.

Samvelyan, M., Rashid, T., Schroeder de Witt, C., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., & Whiteson, S. (2019). *The starcraft multi-agent challenge*. *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2186–2188.

Sartoretti, G., Paivine, W., Shi, Y., Wu, Y., & Choset, H. (2019). *Distributed learning of decentralized control policies for articulated mobile robots*. *IEEE Transactions on Robotics*, *35*(5), 1109–1122.

Scassellati, B., Boccanfuso, L., Huang, C.-M., Mademtzi, M., Qin, M., Salomons, N., Ventola, P., & Shic, F. (2018). *Improving social skills in children with asd using a long-term, in-home social robot*. *Science Robotics*, *3*(21).

Schroeder de Witt, C., Gupta, T., Makoviichuk, D., Makoviychuk, V., Torr, P. H. S., Sun, M., & Whiteson, S. (2020). *Is independent learning all you need in the starcraft multi-agent challenge?* [arxiv:2011.09533].

Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2016). *High-dimensional continuous control using generalized advantage estimation* (Y. Bengio & Y. LeCun, Eds.).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms*.

Shapley, L. S. (1953). *A value for n-person games*. *Contributions to the Theory of Games*, *2*(28), 307–317.

Shih, A., Sawhney, A., Kondic, J., Ermon, S., & Sadigh, D. (2021). *On the critical role of conventions in adaptive human-ai collaboration*. *International Conference on Learning Representations*.

Shoham, Y., Powers, R., & Grenager, T. (2007). *If multi-agent learning is the answer, what is the question?* [Foundations of Multi-Agent Learning]. *Artificial Intelligence*, *171*(7), 365–377.

Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. *3rd International Conference on Learning Representations (ICLR 2015)*, 1–14.

Singh, A., Jain, T., & Sukhbaatar, S. (2019). *Learning when to communicate at scale in multiagent cooperative and competitive tasks*. *International Conference on Learning Representations*.

Singh, S., Mahjoub, O., de Kock, R., Khlifi, W., Vall, A., Tessera, K.-a., & Pretorius, A. (2023). *How much can change in a year? revisiting evaluation in multi-agent reinforcement learning* [arxiv:2312.08463].

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., & Yi, Y. (2019). *Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning*. *Proceedings of the 36th International Conference on Machine Learning*, *PMLR 97*, 5887–5896.

Stone, P., Kaminka, G., Kraus, S., & Rosenschein, J. (2010). *Ad hoc autonomous agent teams: Collaboration without pre-coordination*. *Proceedings of the AAAI Conference on Artificial Intelligence*, *24*(1), 1504–1509.

Stone, P., & Veloso, M. (2000). *Multiagent systems: A survey from a machine learning perspective*. *Autonomous Robots*, *8*(3), 345–383.

Strouse, D., McKee, K., Botvinick, M., Hughes, E., & Everett, R. (2021). *Collaborating with humans without human data*. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (pp. 14502–14515, Vol. 34). Curran Associates, Inc.

Sukhbaatar, S., Szlam, A., & Fergus, R. (2016). *Learning multiagent communication with backpropagation*. *Advances in Neural Information Processing Systems*, 2244–2252.

Sun, W.-F., Lee, C.-K., See, S., & Lee, C.-Y. (2023). *A unified framework for factorizing distributional value functions for multi-agent reinforcement learning*. *Journal of Machine Learning Research*, *24*(220), 1–32.

Sünderhauf, N., Brock, O., Scheirer, W., Hadsell, R., Fox, D., Leitner, J., Upcroft, B., Abbeel, P., Burgard, W., Milford, M., & Corke, P. (2018). *The limits and potentials of deep learning for robotics*. *The International Journal of Robotics Research*, *37*(4-5), 405–420.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., & Graepel, T. (2018). *Value-decomposition networks for cooperative multi-agent learning based on team reward*. *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2085–2087.

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Aru, J., & Vicente, R. (2017). *Multiagent cooperation and competition with deep reinforcement learning*.

Tan, M. (1993). *Multi-agent reinforcement learning: Independent versus cooperative agents*. *Proceedings of the Tenth International Conference on Machine Learning*, 330–337.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). *Domain randomization for transferring deep neural networks from simulation to the real world*. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–30.

Toquebiau, M., Bredeche, N., Benamar, F., & Jun, J.-Y. (2024). *Joint intrinsic motivation for coordinated exploration in multi-agent deep reinforcement learning*. *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 2522–2524.

Toquebiau, M., Jun, J.-Y., Benamar, F., & Bredeche, N. (2025). *Towards language-augmented multi-agent deep reinforcement learning*. *28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy – Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, *413*.

Tseng, S.-H., Chao, Y., Lin, C., & Fu, L.-C. (2016). *Service robots: System design for tracking people through data fusion and initiating interaction with the human group by inferring social situations*. *Robotics and Autonomous Systems*, *83*, 188–202.

Vinitsky, E., Lichtlé, N., Yang, X., Amos, B., & Foerster, J. (2022). *Nocturne: A scalable driving benchmark for bringing multi-agent learning one step closer to the real world*. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 3962–3974, Vol. 35). Curran Associates, Inc.

Wang, J., Ren, Z., Han, B., Ye, J., & Zhang, C. (2021). *Towards understanding cooperative multi-agent q-learning with value factorization*. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (pp. 29142–29155, Vol. 34). Curran Associates, Inc.

Wang, J., Ren, Z., Liu, T., Yu, Y., & Zhang, C. (2021). *Qplex: Duplex dueling multi-agent q-learning*. *9th International Conference on Learning Representations*.

Wang, J., Zhang, Y., Gu, Y., & Kim, T.-K. (2022). *Shaq: Incorporating shapley value theory into multi-agent q-learning*. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 5941–5954, Vol. 35). Curran Associates, Inc.

Wang, J., Zhang, Y., Kim, T.-K., & Gu, Y. (2020). *Shapley q-value: A local reward approach to solve global reward games*. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 7285–7292.

Wang, R. E., Everett, M., & How, J. P. (2020). *R-maddpg for partially observable environments and limited communication*. *Reinforcement Learning for Real Life (RL4RealLife) Workshop in the 36th International Conference on Machine Learning, Long Beach, California, USA, 2019*.

Wang, R., He, X., Yu, R., Qiu, W., An, B., & Rabinovich, Z. (2020). *Learning efficient multi-agent communication: An information bottleneck approach*. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 9908–9918, Vol. 119). PMLR.

Wiegand, R. P. (2003). *An analysis of cooperative coevolutionary algorithms* [Doctoral dissertation, George Mason University].

Wolpert, D. H., & Tumer, K. (1999). *An introduction to collective intelligence* (tech. rep.) (NASA-ARC-IC-99-63). NASA.

Wolpert, D. H., & Tumer, K. (2002). *Optimal payoff functions for members of collectives*. In F. Schweitzer (Ed.), *Modeling complexity in economic and social systems* (pp. 355–369). World Scientific Publishing Co. Pte. Ltd.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John wiley & sons.

Xie, A., Losey, D. P., Tolsma, R., Finn, C., & Sadigh, D. (2021). *Learning latent representations to influence multi-agent interaction*. In J. Kober, F. Ramos, & C. Tomlin (Eds.), *Proceedings of the 2020 conference on robot learning* (pp. 575–588, Vol. 155). PMLR.

Xu, Z., Zhang, B., li dapeng, d., Zhou, G., Zhang, Z., & Fan, G. (2023). *Dual self-awareness value decomposition framework without individual global max for cooperative marl*. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (pp. 73898–73918, Vol. 36). Curran Associates, Inc.

Yan, X., Guo, J., Lou, X., Wang, J., Zhang, H., & Du, Y. (2023). *An efficient end-to-end training approach for zero-shot human-ai coordination*. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (pp. 2636–2658, Vol. 36). Curran Associates, Inc.

Yang, Y., Hao, J., Liao, B., Shao, K., Chen, G., Liu, W., & Tang, H. (2020). *Qatten: A general framework for cooperative multiagent reinforcement learning* [arxiv:2002.03939].

Yu, C., Gao, J., Liu, W., Xu, B., Tang, H., Yang, J., Wang, Y., & Wu, Y. (2023). *Learning zero-shot cooperation with humans, assuming humans are biased*. *The 11th International Conference on Learning Representations*.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., & Wu, Y. (2021a). *Benchmarking multi-agent deep reinforcement learning algorithms*.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., & Wu, Y. (2021b). *The surprising effectiveness of ppo in cooperative, multi-agent games* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh, Eds.). *35*, 24611–24624.

Zhang, S. Q., Zhang, Q., & Lin, J. (2019). *Efficient communication in multi-agent reinforcement learning via variance based control*. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

Zhao, R., Song, J., Yuan, Y., Hu, H., Gao, Y., Wu, Y., Sun, Z., & Yang, W. (2023). *Maximum entropy population-based training for zero-shot human-ai coordination*. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(5), 6145–6153.

Zhou, H., Lan, T., & Aggarwal, V. (2022). *Pac: Assisted value factorization with counterfactual predictions in multi-agent reinforcement learning*. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 15757–15769, Vol. 35). Curran Associates, Inc.

Zhou, M., Liu, Z., Sui, P., Li, Y., & Chung, Y. Y. (2020). *Learning implicit credit assignment for cooperative multi-agent reinforcement learning*. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 11853–11864, Vol. 33). Curran Associates, Inc.

Zhu, C., Dastani, M., & Wang, S. (2024). *A survey of multi-agent deep reinforcement learning with communication*. *Autonomous Agents and Multi-Agent Systems*, *38*(1).

Ziemke, T. (2003). *What's that thing called embodiment? Proceedings of the 25th Annual Cognitive Science Society*.

Złotowski, J., Proudfoot, D., Yogeeswaran, K., & Bartneck, C. (2014). *Anthropomorphism: Opportunities and challenges in human–robot interaction*. *International Journal of Social Robotics*, *7*(3), 347–360.