An Organizationally-Oriented Approach to Enhancing Explainability and Control in Multi-Agent Reinforcement Learning

Julien Soulé
Univ. Grenoble Alpes
Valence, France
julien.soule@lcis.grenoble-inp.fr

Jean-Paul Jamont
Univ. Grenoble Alpes
Valence, France
jean-paul.jamont@lcis.grenoble-inp.fr

Michel Occello
Univ. Grenoble Alpes
Valence, France
michel.occello@lcis.grenoble-inp.fr

Louis-Marie Traonouez
Thales Land and Air Systems, BU IAS
Rennes, France
louis-marie.traonouez@thalesgroup.com

Paul Théron
AICA IWG
La Guillermie, France
paul.theron@orange.fr

Abstract

Multi-Agent Reinforcement Learning can lead to the development of collaborative agent behaviors that show similarities with organizational concepts. Pushing forward this perspective, we introduce a novel framework that explicitly incorporates organizational roles and goals from the $\mathcal{MOISE}^+$ model into the MARL process, guiding agents to satisfy corresponding organizational constraints. By structuring training with roles and goals, we aim to enhance both the explainability and control of agent behaviors at the organizational level, whereas much of the literature primarily focuses on individual agents. Additionally, our framework includes a post-training analysis method to infer implicit roles and goals, offering insights into emergent agent behaviors. This framework has been applied across various MARL environments and algorithms, demonstrating coherence between predefined organizational specifications and those inferred from trained agents.

Keywords

Multi-Agent Reinforcement Learning; Organizational Explainability; Organizational Control

ACM Reference Format:
Julien Soulé, Jean-Paul Jamont, Michel Occello, Louis-Marie Traonouez, and Paul Théron. 2025. An Organizationally-Oriented Approach to Enhancing Explainability and Control in Multi-Agent Reinforcement Learning. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

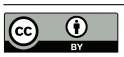
1 Introduction

Multi-Agent Reinforcement Learning (MARL) enables the discovery of a joint policy that controls agents' behaviors so they can achieve a global goal within a specific environment. This joint policy not only dictates the individual actions of agents but also manages their interactions with one another, and potentially with all other agents, without any preconceived notion of a predefined organization.

In environments that require social interaction among agents to optimally achieve the global goal, agents may converge in such a way that they exhibit recurring sets of similar behaviors across different testing episodes. These distinct sets of behaviors can demonstrate properties of specialization, complementarity, and stability, making them akin to implicit roles. Moreover, the trajectories of agents assuming these "implicit" roles may display similarities, such as recurrent observations at the end of each episode. These recurring patterns in agent histories can be interpreted as "implicit" goals, suggesting that agents may aim to pursue these as intermediate goals before reaching the global goal. These implicit roles and implicit goals form the foundation of an "implicit" structural and functional organization as defined in $\mathcal{MOISE}^+$ [14].

However, it would be misleading to assume that all trained agents in any environment can be faithfully compared to a structural and functional organization. Indeed, we can interpret the behaviors of trained agents concerning their similarity to the potential vision of an implicit structural and functional organization, which we define as **organizational fit**. While evaluating organizational fit would be useful to assess to what extent trained agents can naturally be explained as roles and goals, one could also consider the reverse approach. By guiding or encouraging agents to converge towards structural and functional organizations with higher organizational fit, we aim to enhance explainability and control in MARL.

Building on these assumptions, this paper aims to further explore two key aspects: i) The **evaluation of organizational fit**, which seeks to measure how closely a joint policy aligns with a structural and functional organization. A significant challenge here is to understand under what conditions agents can be considered to form a structural and functional organization, given constraints imposed by the environment, goals, and other optional factors. Existing literature often addresses policy evaluation in terms of roles or goals [15, 27, 29], but these works generally lack a systematic and comprehensive approach. Current methods offer few clear tools



Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

for quantitatively and qualitatively measuring this organizational fit.; ii) The **control of organizational fit**, which aims to guide agents towards policies that conform to a structural and functional organization through user-defined constraints or incentives that implement roles and goals. The primary challenges include reducing the policy search space, improving convergence, and ensuring compliance with safety constraints. Existing approaches in this field often fall short in terms of enabling users to easily define and manage the application of organizational specifications in a practical manner within a standard MARL framework, without relying on paradigms such as Hierarchical Reinforcement Learning (HRL).

We introduce the **MOISE+MARL** framework, which integrates the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) MARL framework with the $\mathcal{M}OISE^{+}$ [14] organizational model through proposed relationships. This framework allows users to manually define the logic of a role or a goal by relying on trajectory-based patterns to describe the expected behavior of an agent that has adopted a goal or mission. Once configured, they allow users to apply a role to an agent, adding constraints that automatically influence agents' policies by dynamically updating both the action space and reshaping the reward function. This framework also includes a method called **Trajectory-based Evaluation in MOISE+MARL** (TEMM), which uses unsupervised learning techniques to generalize implicit roles and implicit missions from observed trajectories across multiple test episodes. By measuring the gap between inferred implicit organizational specifications and actual behaviors, this method allows for a quantitative assessment of organizational fit. It is worth noting that unlike hierarchical reinforcement learning, which decomposes tasks into subtasks [18, 20, 23], our approach relies on explicit organizational roles and missions to guide agent coordination externally.

We evaluated the MOISE+MARL framework in the following scenarios: i) Four distinct environments, each expected to result in the training of joint policies with different implicit organizations, to assess the generalizability of MOISE+ MARL's applicability; ii) Four MARL algorithms from the several families to assess their suitability with MOISE+ MARL during training and post-analysis; iii) Four sets of organizational specifications, one for each environment, to constrain agents in a manner that either enforces conformity intended for both manual and quantitative evaluation.

In all environments, we observed that agents having adopted roles do behave as expected according to their roles in a correlated way with a quantitative measure of the organizational fit by TEMM. The roles and missions inferred by TEMM closely align with the predefined specifications, demonstrating the internal consistency of MOISE+MARL, as the policy modifications introduced by organizational specifications are effectively captured by TEMM. The results also indicate that policy-based and actor-critic algorithms are particularly well-suited for guiding agents towards stable policies. This stability allows agents to maintain consistent and coherent behaviors across episodes, which is essential for TEMM's generation of a stable implicit organization. In contrast, value-based algorithms showed greater variability in agent behaviors.

The rest of the paper is organized as follows: Section 2 presents works relative to evaluating and controlling organizational fit. Section 3 introduces the MOISE+MARL framework. Section 4 describes the TEMM method. Section 5 describes the experimental protocol, particularly the environments and MARL algorithms. Section 6 presents the experimental results. Finally, Section 7 discusses and concludes on the evaluation and control of organizational fit.

2 Related works

This section explores works related to organizational fit, as framed by the two core issues introduced.

2.1 Evaluating organizational fit

Some works may be related to role or goal inference regarding the need to compute organizational fit or close concepts. Wilson et al. [28] develop a method for transferring roles in Multi-Agent MDPs, which helps agents adapt by transferring roles across different environments. However, their model lacks the role abstraction as it focuses on specific, task-related roles. Berenji and Vengerov [5] investigate coordination and role inference in UAV missions, enhancing cooperation through modeling agent dependencies. While useful for cooperation, their approach remains task-specific and does not provide the implicit role computation needed for organizational fit. Yusuf and Baber [31] use inferential reasoning and Bayesian methods to facilitate task coordination among diverse agents. Though effective in dynamic coordination, their framework lacks role abstraction and does not measure alignment with an broader organizational structure either. Serrino et al. [24] examine dynamic role inference in social settings, where agents deduce roles through interactions. While they enable flexible role understanding, their approach focuses on immediate operational roles rather than implicit roles that align with organizational models.

While some works explore organizational concepts in MARL, none explicitly address the computation of organizational alignment as we define it. Our concept of organizational fit requires a framework that assesses alignment with implicit goals.

2.2 Controlling organizational fit

Controlling organizational fit involves aligning the agents' policies with a predefined organization, often using constraints or incentives. Achiam et al. [1] introduce CPO, adjusting policies with safety constraints while maximizing rewards. MOISE+MARL, however, introduces constraints beyond safety to shape behavior toward organizational expectations by externally guiding agent learning. Ray et al. [22] use Lagrange multipliers to integrate constraints into the reward function, balancing reward and constraint adherence. MOISE+MARL extends this by dynamically modifying the action space to enforce constraint adherence at various levels, offering flexible control over agent behaviors. Safe exploration ensures agents learn while adhering to safety constraints. Garcia et al. [11] overview methods for maintaining safe exploration, and Alshiekh et al. [4] propose shielding to block unsafe actions. MOISE+MARL goes further by using constraints to guide agents toward behaviors that align with organizational roles. HRL breaks tasks into subtasks, aligning with organizational hierarchies. Ghavamzadeh et al. [12]

illustrate that HRL can improve coordination. MOISE+MARL constrains MARL externally, offering a modular granularity and generating refined behaviors under organizational constraints. Controlling Communication and Coordination is essential for ensuring organizational fit, especially in large-scale systems. Foerster et al. [10] propose decentralized coordination through shared knowledge, allowing agents to operate without centralized control.

Unlike HRL, the MOISE+MARL framework stands out for incorporating external organizational constraints that influence agents within a standard MARL framework, enabling modular granularity. Unlike Shielding or CPO, which typically focus on safety constraints, MOISE+MARL goes further by relying on actions and reward modifications to align with roles.

3 The MOISE+MARL framework

This section introduces the formalism used to describe the functioning framework of the MOISE+MARL framework.

3.1 Markov framework for MARL

To apply MARL techniques, we rely on the *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP) [19]. Dec-POMDPs naturally model decentralized multi-agent coordination under partial observability, making them well suited for integrating organizational constraints. Unlike *Partially Observable Stochastic Games* (POSG), the Dec-POMDP allows for a common reward function for agents, which promotes collaboration [6].

A Dec-POMDP $d \in D$ (where D is the set of Dec-POMDPs) is defined as a 7-tuple $d = \langle S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma\rangle$, where $S = \{s_1, \ldots, s_{|S|}\}$ is the set of possible states; $A_i = \{a_1^i, \ldots, a_{|A_i|}^i\}$ is the set of possible actions for agent i; T represents the set of transition probabilities, with $T(s, a, s') = \mathbb{P}(s'|s, a)$ as the probability of transitioning from state s to state s' following action a; $R : S \times A \times S \to \mathbb{R}$ is the reward function, assigning a reward based on the initial state, the action taken, and the resulting state; $\Omega_i = \{o_1^i, \ldots, o_{|\Omega_i|}^i\}$ is the set of possible observations for agent i; O represents the set of observation probabilities, where $O(s', a, o) = \mathbb{P}(o|s', a)$ is the probability of obtaining observation o after performing action a and reaching state s'; and $\gamma \in [0, 1]$ is the discount factor

The following formalism is used with MOISE+MARL to solve the Dec-POMDP [3, 6]: $\mathcal{A}$ represents the set of n **agents**; Π denotes the set of **policies**, where a policy $\pi \in \Pi, \pi : \Omega \to A$ deterministically maps an observation to an action, representing the agent's internal strategy; Π_{joint} represents the set of **joint policies**, with a joint policy $\pi_{joint} \in \Pi_{joint}, \pi_{joint} : \Omega^n \to A^n = \Pi^n$, which selects an action for each agent based on their respective observations, acting as a collection of policies used by agents within a team; H is the set of **histories**, where a history (or trajectory) over $z \in \mathbb{N}$ steps (typically the maximum number of steps in an episode) is represented as the z-tuple $h = \langle\langle \omega_k, a_k\rangle | k \leq z, \omega \in \Omega, a \in A\rangle$, capturing successive observations and actions; H_{joint} stands for the set of **joint histories**, with a joint history $h_{joint} \in H_{joint}$ over z steps defined as the set of agent histories: $h_{joint} = \{h_1, h_2, \ldots, h_n\}$; and finally, $V_{joint}(\pi_{joint}) : \Pi_{joint} \to \mathbb{R}$ denotes the **expected cumulative reward** over a finite horizon (assuming $\gamma < 1$ or if the number of steps in an episode is finite), where π_{joint} represents the joint policy for team i, with $\pi_{joint,-i}$ being the joint policies of other teams, considered as fixed.

3.2 The $\mathcal{MOISE}^+$ organizational model

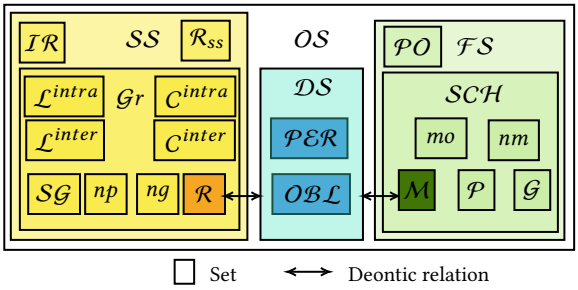


Figure 1: A synthetic view of the $\mathcal{MOISE}^+$ model

As illustrated in Figure 1, $\mathcal{MOISE}^+$ comprises three types of organizational specifications:

Structural Specifications (SS) define how agents are structured, expressed as $\mathcal{SS} = \langle \mathcal{R}, \mathcal{IR}, \mathcal{G}\rangle$. $\mathcal{R}_{ss}$ is the set of roles ($\rho \in \mathcal{R}$) with an inheritance relation $\mathcal{IR}$ where $\rho_1 \sqsubset \rho_2$ if ρ_1 inherits from ρ_2. $\mathcal{GR}$ includes groups $\langle \mathcal{R}, \mathcal{SG}, \mathcal{L}^{intra}, \mathcal{L}^{inter}, \mathcal{C}^{intra}, \mathcal{C}^{inter}, np, ng\rangle$. Links ($\mathcal{L}$) define connections between roles: acquaintance, communication, or authority. Compatibilities $\mathcal{C}$ denote roles that agents can play together. Intra- and inter-group links and compatibilities are shown by $\mathcal{L}^{intra}$, $\mathcal{L}^{inter}$, $\mathcal{C}^{intra}$, and $\mathcal{C}^{inter}$, with np and ng defining role and subgroup counts.

Functional Specifications (FS) describe the agents' goals, represented as $\mathcal{FS} = \langle \mathcal{SCH}, \mathcal{PO}\rangle$. The social scheme $\mathcal{SCH}$ includes global goals $\mathcal{G}$, missions $\mathcal{M}$, and plans $\mathcal{P}$ that organize goals in a tree structure. Plans link goals with an operator (op) indicating sequence, choice, or parallel completion. Missions map to goal sets (mo), and agent counts per mission are specified by nm. Preferences $\mathcal{PO}$ indicate which missions agents prefer, denoted as $m_1 \prec m_2$.

Deontic Specifications (DS) indicate the relationship between roles goals, given by $\mathcal{DS} = \langle \mathcal{OBL}, \mathcal{PER}\rangle$. Time constraints $\mathcal{TC}$ set periods for permissions or obligations (*Any* for any time). Obligations ($\mathcal{OBL}$) require agents in role ρ_a to undertake mission m at times tc, while permissions ($\mathcal{PER}$) allow it. The rds function maps roles to their deontic specifications as $\langle tc, y, m\rangle$ where y distinguishes permission (0) from obligation (1).

Organizational specifications applied to agents are roles and goals (as missions) through permissions or obligations. Indeed, the other structural specifications such as compatibilities or links are inherent to roles. Similarly, we consider that the goals, the missions, and their mapping (mo) are enough to also link all of the other functional specifications such as plans, cardinalities, or preference orders. Consequently, we consider it is sufficient to take into account roles, missions (goal and mapping) and permissions/obligations when linking $\mathcal{MOISE}^+$ with Dec-POMDP.

Definition 1 **Sate-Value function adapted to constraint guides in AEC mode:**

$$V^{\pi^j}(s_t) = \sum_{\substack{a_t \in A \text{ if } rn()<ch_t, \\ a_t \in A_t \text{ else}}} \pi_i(a_t|\omega_t) \sum_{s_{t+1} \in S} T(s_{t+1}|s_t, a_t)[R(s_t, a_t, s_{t+1}) + \sum_{m \in \mathcal{M}_i} v_m(t) \frac{grg_m(h_{t+1})}{1-p+\epsilon} + (1-ch_t) \times rrg(\omega_t, a_{t+1}) + V^{\pi^j_{i+1 \bmod n}}(s_{t+1})]$$

With $rag(h_t, \omega_t) = A_t \times \mathbb{R}$, $\langle a_t, ch_t \rangle \in A_t \times \mathbb{R}$; and $rn : \emptyset \rightarrow [0, 1[$, a uniform random function
With $\omega_t = O(\omega_t|s_t, a_t)$; $h_t = \{h_0 = \langle\rangle, h_{t+1} = \langle h_t, \langle \omega_{t+1}, a_{t+1} \rangle\rangle\}$; $grg_m(h) = \sum_{(grg_i, w_i) \in mo(m)} w_i \times grg_i(h)$; $\epsilon \in \mathbb{R}_{>0}$;
$v_m(t) = \{1 \text{ if } t \in t_c \text{ ; else } 0\}$; and $\mathcal{M}_i = \{m_j | \langle ar(i), m_j, t_c, p \rangle \in \mathcal{M}\}$

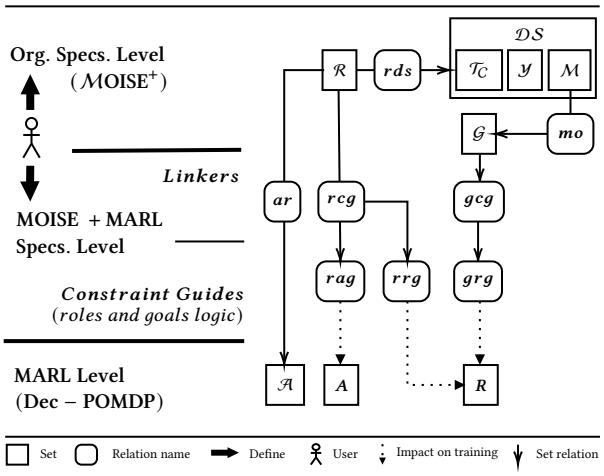


Figure 2: A minimal view of the MOISE+MARL framework: Users first define $\mathcal{MOISE}^+$ specifications, which include roles ($\mathcal{R}$) and missions ($\mathcal{M}$), both associated through *rds*. They then create MOISE+MARL specifications by first defining Constraint guides such as *rag* and *rrg* to specify role logic, and *grg* for goal logic. Next, Linkers are used to connect agents with roles through *ar* and to link the logic of the constraint guides to the defined $\mathcal{MOISE}^+$ specifications. Once this is set up, roles can be assigned to agents, and the MARL framework updates accordingly during training.

3.3 Linking $\mathcal{MOISE}^+$ with MARL

We identified the *AGR* [8] (Agent Group Role) and the $\mathcal{MOISE}^+$ [14] organizational models. Unlike AGR which is an informal framework introducing roles according to groups, $\mathcal{MOISE}^+$ provides a more detailed and flexible description of the structures and functions of a MAS, easing a formal description of agents' policies in MARL.

The **Constraint Guides** are three new relations introduced to describe the logic of the roles and goals of $\mathcal{MOISE}^+$ in the Dec-POMDP formalism: i) **Role Action Guide** $rag : H \times \Omega \rightarrow \mathcal{P}(A \times \mathbb{R})$, the relation that models a role as a set of rules which, for each pair consisting of a history $h \in H$ and an observation received by the agent $\omega \in \Omega$, associates expected actions $A \in \mathcal{P}(A)$ each associated with a constraint hardness $ch \in [0, 1]$ ($ch = 1$ by default). By restricting the choice of the next action among those authorized, the agent is forced to adhere to the expected behavior of the role; ii) **Role Reward Guide** $rrg : H \times \Omega \times A \rightarrow \mathbb{R} = \{r_m \text{ if } a \notin A_\omega, rag(h, \omega) = A_\omega \times \mathbb{R}, h \in H;$ else $0\}$, the relation that models a role by adding a penalty r_m to the global reward if the last action chosen by the agent $a \in A$ is not authorized. This is intended to encourage the agent to adhere to the expected behavior of a role; iii) **Goal Reward Guide** $grg : H \rightarrow \mathbb{R}$, the relation that models a goal as a soft constraint by adding a bonus $r_b \in \mathbb{R}$ to the global reward if the agent's history $h \in H$ contains a characteristic sub-sequence $h_g \in H_g$ of the goal, encouraging the agent to reach it.

Finally, we introduce the **Linkers** to link the $\mathcal{MOISE}^+$ organizational specifications with constraint guides and agents: i) **Agent to Role** $ar : \mathcal{A} \rightarrow \mathcal{R}$, the bijective relation linking an agent to a role;; ii) **Role to Constraint Guide** $rcg : \mathcal{R} \rightarrow rag \cup rrg$, the relation associating each $\mathcal{MOISE}^+$ role to a *rag* or *rrg* relation, forcing/encouraging the agent to follow the expected actions for the role $\rho \in \mathcal{R}$;; iii) **Goal to Constraint Guide** $gcg : \mathcal{G} \rightarrow grg$, the relation linking goals to *grg* relations, representing goals as rewards in MARL.

Resolving the MOISE+MARL problem involves finding a joint policy $\pi^j = \{\pi^j_0, \pi^j_1 \dots \pi^j_n\}$ that maximizes the state-value function V^{π^j} (or reaches a minimum threshold), which represents the expected cumulative reward starting from an initial state $s \in S$ and following the joint policy π^j, applying successive joint actions $a^j \in A^n$ under additional constraint guides. The state-value is described in the case where agents act sequentially and cyclically (Agent Environment Cycle - AEC mode) in Definition 1, adapting its definition for roles (in red) and missions (in blue), impacting the action space and reward. Figure 2 illustrates the links between $\mathcal{MOISE}^+$ and Dec-POMDP via the MOISE+MARL framework.

At any time $t \in \mathbb{N}$ (initially $t = 0$), the agent $i = t \bmod n$ is constrained to a role $\rho_i = ar(i)$. For each temporally valid deontic specification $d_i = rds(\rho_i) = \langle tc_i, y_i, m_i \rangle$, the agent is permitted (if $y_i = 0$) or obligated (if $y_i = 1$) to commit in mission $m_i \in \mathcal{M}, \mathcal{G}_{m_i} = mo(m_i)$, and $n \in \mathbb{N}$ the number of agents. First, based on the received observation ω_t, the agent must choose an action either: within the expected actions of the role A_t if a random value is below the role constraint hardness ch_t; or within the set of all actions A otherwise. If $ch_t = 1$, the role is strongly constrained for the agent and weakly otherwise. Then, the action is applied to the current state s_t to transition to the next state s_{t+1}, generate the next observation ω_{t+1}, and yield a reward. The reward is the sum of the global reward with penalties and bonuses obtained from the organizational specifications: i) the sum of the bonuses for goals associated with each temporally valid mission (via Goal Reward Guides), weighted by the associated value ($\frac{1}{1-p+\epsilon}$); ii) the penalty associated with the role (via "Role Reward Guides") weighted by the role constraint hardness. Finally, the cumulative reward calculation continues in the next state $s_{t+1} \in S$ with the next agent $(i+1) \bmod n$.

3.4 Easying constraint guides implementation

Since roles, goals, and missions as simple labels, their definition is assumed. However, implementing a *rag*, *rrg*, or *grg* relation requires defining a potentially large number of histories, possibly redundant. Therefore, an extensional definition of a set of histories can be tedious. Moreover, the logic of all constraint guides takes the agent trajectory as input to determine whether the trajectory belongs to a predefined history set. For example, a *rag* relation can be seen as determining the next expected actions depending on whether the trajectory belongs to a given set and the new observation received.

A first approach is to let users develop their constraint guides in an intensional way with custom logic (such as a script code) in order to analyse history and compute the output in a manageable way. In that case, the relation $b_g : H \rightarrow \{0, 1\}$ formalizes how users propose to determine whether a history belongs to a predefined set H_g. To help implement this relation, we propose a **Trajectory-based Pattern** (TP) inspired by Natural Language Processing, denoted $p \in P$, as a way to define a set of histories in an intensional way.

A TP implies that any considered real observation or action is known and mapped to a label $l \in L$ (through $l : \Omega \cup A \rightarrow L$) to be conveniently managed. A TP $p \in P$ is defined as follows: p is: either a "leaf sequence" denoted as a couple of history-cardinality $s_l = \langle h, \{c_min, c_max\}\rangle$ (where $h \in H, c_{min} \in \mathbb{N}, c_{max} \in \mathbb{N} \cup " * "$); or a "node sequence" denoted as a couple of a tuple of concrete sequences and cardinality $s_n = \langle\langle s_{l_1}, s_{l_1} \dots\rangle, \{c_min, c_max\}\rangle$. For example, the pattern $p =$ "$[o_1, a_1, [o_2, a_2]\langle 0, 2\rangle]\langle 1, *\rangle$" can be formalized as the node sequence $\langle\langle\langle o_1, a_1\rangle, \langle 1, 1\rangle\rangle, \langle\langle o_2, a_2\rangle, \langle 0, 2\rangle\rangle\rangle\langle 1, "*"\rangle$, indicating the set of histories H_p containing at least once the sub-sequence consisting of a first pair $\langle o_1, a_1\rangle$ and then at most two repetitions of the pair $\langle o_2, a_2\rangle$. The relation b_g then becomes $b_g(h) = m(p_g, h),$ with $m : P \times H \rightarrow \{0, 1\}$ indicating if a history $h \in H$ matches a history pattern $p \in P$ describing a history set H_g.

4 The TEMM method

As presented in Section 2, we were unable to identify any available method that fully meets our requirements for determining implicit roles, implicit goals, or organizational fit. Therefore, we propose the **Trajectory-based Evaluation in MOISE+MARL** (TEMM) method for automatic inference and evaluation of roles and missions. TEMM uses unsupervised learning techniques to generalize roles and missions from the set of collected trajectories over multiple test episodes. By measuring the gap between inferred implicit organizational specifications and actual behaviors, we can also quantify the organizational fit as to how well a policy conforms to the inferred implicit organizational specifications.

TEMM is based on proposed definitions for each $\mathcal{M}OISE^+$ organizational specification regarding joint-histories or other organizational specifications, using specific unsupervised lea-rning techniques to infer them progressively. Here, we provide an informal description of the method [1].

1) Inferring roles and their inheritance We introduce that a role ρ is defined as a policy whose associated agents' histories all contain a Common Longest Sequence (CLS). We introduce that a role ρ_2 inherits from ρ_1 if the CLS of histories associated with ρ_2 is also contained within that of ρ_1. Based on these definitions, TEMM uses a "hierarchical clustering" technique to find the CLSs among agent histories. The results can be represented as a dendrogram, allowing inferring implicit roles and inheritance relationships, their respective relationships with histories. We measure the gap between current agents' sequence and inferred implicit roles' sequences, as the "structural organizational fit".

2) Inferring goals, plans, and missions We introduce that a goal is a set of common joint-observation reached by following the histories of successful agents. For each joint-history, TEMM calculates the joint-observation transition graph, which is then merged into a general graph. By measuring the distance between two vectorized joint-observations with K-means, we can find trajectory clusters that some agents may follow. Then, we sample some sets of joint-observations for each trajectory as implicit goals. For example, we can select the narrowest set of joint-observations where agents seem to collectively transition at a given time to reach their goal. Otherwise, balanced sampling on low-variance trajectories could be performed. Knowing which trajectory a goal belongs to, TEMM infers plans based solely on choices and sequences.

We introduce that a mission is the set of goals that one or more agents are accomplishing. Knowing the shared goals achieved by the agents, TEMM determines representative goal sets as missions. By measuring the distance between inferred implicit goals which joint-observations with current agents' joint-observation, we compute the "structural organizational fit".

3) Inferring obligations and permissions We introduce that an obligation is when an agent playing the role ρ fulfills the goals of a mission and no others during certain time constraints, while permission is when the agent playing the role ρ may fulfill other goals during specific time constraints. TEMM determines which agents are associated with which mission and whether they are restricted to certain missions, making them obligations, or if they have permission. Having already computed structural organizational fit and functional organizational fit, the organizational fit is the sum of these two values.

Overall, the K-mean and hierarchical clustering techniques require manual configuration to obtain roles and goals, avoiding introducing perturbations that could lead to determining false organizational specifications. Despite this, the method recommends thoroughly understanding the obtained roles and goals to manually identify and remove any remaining perturbations.

5 Experimental framework

This section details the experimental framework used to evaluate the MOISE+MARL framework.

5.1 Implementing MOISE+MARL

We have developed an implementation of the MOISE+MARL framework called "MMA" [1] (MOISE+MARL API), which is a Python API that integrates all theoretical sets and relations to minimize user interactions. MMA uses an Object-oriented approach, structuring the $\mathcal{M}OISE^+$ model as nested data classes, with the "Moise" class

[1] Additional details, developed code, datasets containing all the hyperparameters and details of the organizational specifications are available at https://github.com/julien6/MOISE-MARL

at the root, enabling users to define organizational specifications, such as roles, goals, and permissions.

To support Dec-POMDP environments, we utilized the *PettingZoo* library [26], which provides a standard API for multi-agent systems and ensures interoperability across various environments, similar to the Gymnasium framework [16]. MMA incorporates a dictionary for observation/action label mapping (l), which users can customize, and it also supports Trajectory Patterns (TPs) to facilitate pattern definition and matching.

Each type of constraint guide, like *rag*, *rrg*, and *grg*, is implemented as a separate class. Users can define these guides with custom functions or JSON rules; for example, *rag* can be instantiated by associating a ⟨TP, last observation⟩ pair with expected actions, while *grg* can apply bonuses based on specific TPs. The global "MMA" class integrates these guides with user-defined relations, such as linking an agent to a role (*ar*) or associating a role with *rrg* and *rag*, incorporating the organizational specifications defined in the $\mathcal{M}OISE^+$ structure.

Once set up, the MMA object is used to encapsulate the environment with a *PettingZoo* wrapper. This wrapper applies action masks and modifies rewards at each step, ensuring that agents adhere to the organizational specifications throughout training. MMA also integrates *MARLlib* [13], which provides access to state-of-the-art MARL algorithms, enabling training to be run on a high-performance computing cluster.

After training, the TEMM method is employed, using manually optimized hyperparameters to infer implicit roles and goals through hierarchical clustering and K-means. This analysis generates visual outputs, such as dendrograms for roles and joint-observation transition graphs for goals. The resulting implicit roles and goals can be exported as JSON trajectories, providing a structured view of the inferred organizational behaviors.

5.2 Environments used

We test MOISE+MARL in four different MARL environments, each modeled as a Dec-POMDP simulation scenario. These environments were selected for their diversity in terms of collaboration and resource management. Here is a description of each:

- **Predator-Prey**: A classic environment where several predators must cooperate to capture prey. This environment tests the agents' ability to coordinate their actions to achieve a collective goal[17]
- **Overcooked-AI**: A team cooking game where several agents must collaborate to prepare and serve dishes in increasingly complex kitchens[7]. Agents must manage tasks such as chopping, cooking, assembling, and serving ingredients while optimizing their movements and avoiding obstacles. This environment is ideal for testing coordination and task allocation in dynamic, highly interdependent scenarios, where clear roles (such as "chef," "assistant," "server") can be defined via organizational specifications
- **Warehouse Management**: A proposed environment, where agents must manage a warehouse by coordinating resource deliveries to demand points. Roles and missions here influence agent specialization in specific tasks (transportation of products, inventory management)
- **Cyber-Defense Simulation**: A complex environment simulating network defense against cyberattacks. Agents must identify and counter threats while adhering to strict security rules, thus testing the safety of trained agents[25].

These environments are encapsulable in the PettingZoo API, enabling seamless integration with our MOISE+MARL implementation and facilitating the application of organizational specifications.

5.3 MARL algorithms used

We evaluated our framework with several MARL algorithms : i) **MADDPG (Multi-Agent Deep Deterministic Policy Gradient)** [17]: A centralized learning, decentralized execution algorithm, allowing each agent to have a deterministic policy while using global information during training; ii) **MAPPO (Multi-Agent Proximal Policy Optimization)** [30]: An adapted version of PPO for MAS, optimized for stable joint policy convergence in complex scenarios; iii) **Q-Mix** [21]: A Q-value-based algorithm that learns to combine individual agents' Q-values into a joint value to optimize cooperation; iv) **COMA (Counterfactual Multi-Agent)** [9] An actor-critic algorithm able to estimate the impact of an individual agent's actions on the team's overall reward.

5.4 Organizational specifications

For each environment, we defined a set of organizational specifications. These specifications include roles, missions, as well as permissions and obligations. Here, we give an informal description of these [1]: i) **Predator-Prey**: Predator and prey roles are defined, with each predator having specific goals such as "capture the prey" or "block escape routes."; ii) **Overcooked-AI**: Agents adopt three main roles: chef, assistant, and server. The Chef is responsible for cooking and assembling dishes, the Assistant handles ingredient chopping and supply, and the Server is in charge of delivering dishes to customers. Missions primarily involve preparing and serving a certain number of dishes within a given time.; iii) **Warehouse Management**: Agents adopt roles such as "transporter" and "inventory manager," with missions related to managing logistics flows and optimized delivery.; iv) **Cyber-Defense Simulation**: Agents have network defender roles, each with obligations such as intrusion detection or protecting specific drone swarm ad hoc networks.

5.5 Computing resources and hyperparameters

All experiments were conducted on an academic high-performance computing cluster, utilizing various configurations of GPU nodes. Specifically, we employed nodes equipped with NVIDIA A100 and V100 GPUs, and AMD MI210 GPUs. Each algorithm-environment combination was executed on 5 parallel instances to ensure robust and consistent results. Hyperparameters [1] for each algorithm, including learning rates, discount factors, and exploration rates, were either retrieved from MARLlib data banks or optimized for each environment through a grid search using the *Optuna* tool [2].

5.6 Evaluation metrics and protocol

To measure the policy effectiveness and the impact of organizational specifications, we defined the following metrics: i) **Cumulative Reward**: Measures policy effectiveness in achieving environment goals; ii) **Reward Standard Deviation**: Reflects the stability

of learned policies over episodes; iii) **Convergence Rate**: Indicates the speed at which policies achieve stable performance; iv) **Constraint Violation Rate**: Assesses policy adherence to organizational constraints, critical for safety; v) **Consistency Score**: Evaluates alignment between trained behaviors and organizational specifications; vi) **Robustness Score**: Measures agents' ability to maintain performance under a series of challenging scenarios; vii) **Organizational Fit Level**: Quantifies the organizational fit.

Our protocol compares the *Reference Baseline* (RB) without organizational constraints and the *Organizationally Constrained Baseline* (OB) using MOISE+MARL.

We use the MMA software to establish the RB with no organizational specifications. For each environment, we train agents with each algorithm until rewards converge or a maximum episode limit is reached. We record metrics and select the algorithm that achieves the highest Cumulative Reward as the RB (control scenario without constraints). For the OB, we reset environments and agents, applying pre-defined organizational specifications using MMA so that each agent is assigned a role. We train these agents with the RB's highest-performing algorithm, again until convergence or the episode limit. After training, we compute all metrics, providing a scenario with organizational constraints as the OB.

By comparing the RB and OB, we can validate the impact of MOISE+MARL on organizational fit. First, we check if the agents' behaviors align with the specified roles in the OB. We analyze manually or rely on reliable metrics like Reward Standard Deviation, Convergence Rate, and Robustness Score. If agents behave in ways that align with their roles, then we favor the idea that MOISE+MARL has influenced organizational fit. Therefore, we should observe differences in the Organizational Fit Level metric between RB and OB. We can also push forward a correlation between fully/freely constraining roles and higher/lower Organizational Fit Level. If all of these observations hold, then the Organizational Fit Level may quantify the organizational fit, and the Consistency Score metric may be used to validate the effectiveness of MOISE+MARL in controlling organizational fit when roles are applied.

Finally, we also check the relevance of the $\mathcal{MOISE}^+$ by comparing MOISE+MARL with its AGR equivalent called AGR+MARL which only considers roles and does not explicitly include goals.

6 Results

This section presents and analyzes the experimental results from applying MOISE+MARL across the environments.

6.1 Quantitative organizational fit and consistency

Table 1 summarizes the performance metrics for each environment and the most efficient algorithm under both the RB and OB. Across all environments, the organizational fit metric is significantly higher under the OB, confirming that MOISE+MARL effectively aligns agent behaviors with organizational specifications.

For example, in the **Predator-Prey** environment with **MADDPG**, agents in the OB configuration achieved an organizational fit level of 0.87, which represents a 44% increase compared to the RB (0.43). Similarly, in the **Overcooked-AI** environment, **MAPPO** under the OB reached an organizational fit of 0.91 (an increase of 89% over the RB's 0.48). These improvements are mirrored in the **Warehouse Management** environment with **Q-Mix**, where the organizational fit rose from 0.50 in the RB to 0.90 in the OB, suggesting a MOISE+MARL's consistent effectiveness.

In general, agents constrained with organizational specifications show a lower reward deviation and a higher convergence rate that suggests an impact on their behavior. We manually observed agents' interactions in visualizable environments such as Predator-Prey and verified that trained agents' behaviors do align with the expected behavior of a structural and functional implicit organization. Indeed, the significant variation depending on the application of organizational specifications on agents, and the manually verified alignment of agents with roles suggests that organisational fit level correlates with the organizational fit.

Considering organizational fit level reliable across all environments, the **consistency score** also shows important values with a minimal value of 0.76 for the **Cyber-Defense** environment. This suggests that despite a noisy environment that introduces some disturbance in agents' behavior, the inferred organizational specifications are still close to applied ones.

6.2 Performance and stability across algorithms

The results indicate that policy-based and actor-critic algorithms like **MADDPG** and **MAPPO** benefit substantially from the MOISE+MARL framework, particularly in terms of consistency and stability. For example, **MAPPO** in the **Overcooked-AI** environment saw a reward standard deviation reduction from 15.6 (RB) to 10.4 (OB), reflecting a more stable policy with less behavioral fluctuation. **MADDPG** in **Predator-Prey** also showed a similar pattern, with a standard deviation drop from 21.5 in the RB to 15.2 in the OB, indicating increased reliability.

In contrast, value-based algorithms like **Q-Mix** maintained high performance in cumulative reward but displayed greater variability in consistency. For instance, in **Warehouse Management**, **Q-Mix** achieved a reward standard deviation of 13.8 in the OB, a notable improvement over 18.9 in the RB but still higher than the stability observed in policy-based algorithms. This suggests that while **Q-Mix** is effective for achieving task goals, it may require further tuning for roles with MOISE+MARL to enhance consistency.

6.3 Impact of organizational constraints on policy convergence, robustness and violation rates

Applying organizational constraints resulted in faster convergence rates across all environments. In the **Cyber-Defense** environment, **COMA** with MOISE+MARL converged at a rate of 0.86, compared to 0.70 in the RB. Similar trends were observed in the **Warehouse Management** environment with **Q-Mix**, which showed an improvement from 0.74 in the RB to 0.88 in the OB. This expedited convergence can be attributed to the structured guidance of roles and missions, which narrows the policy search space.

In addition to the presented results where constraint hardness is set to 1, we observed that constraint violation rates were consistently higher when organizational constraints were defined with a lower constraint hardness. In **Overcooked-AI**, **MAPPO** recorded

Table 1: Detailed results for each environment and favored algorithm under both RB and OB.

Env.	Alg.	Org. Spec.	Cum. Rew.	STD	Conv. Rate	Viol. Rate	Cons. Score	Rob. Score	Org. Fit Lvl
Predator-Prey	MADDPG		200.1	21.5	0.65	12.3%	-	0.65	0.43
Predator-Prey	MADDPG	Yes	245.8	15.2	0.85	.0%	0.81	0.83	0.87
Overcooked-AI	MAPPO		348.2	15.6	0.75	7.1%	-	0.71	0.48
Overcooked-AI	MAPPO	Yes	391.2	10.4	0.92	.0%	0.89	0.89	0.91
Warehouse Management	Q-Mix		257.4	18.9	0.74	7.8%	-	0.68	0.50
Warehouse Management	Q-Mix	Yes	307.1	13.8	0.88	.0%	0.88	0.86	0.90
Cyber-Defense	COMA		162.4	17.3	0.70	12.2%	-	0.67	0.45
Cyber-Defense	COMA	Yes	188.9	11.2	0.86	.0%	0.76	0.80	0.83

a null violation rate with a constraint hardness of 1, compared to 7.1% with a constraint hardness of 0. Similarly, in **Warehouse Management**, **Q-Mix** reduced the violation rate from 7.8% to zero as constraint hardness increased. This further supports the framework's effectiveness in enhancing adherence to desired behaviors.

Additionally, we observed a consistent improvement in robustness when organizational specifications were applied to agents. For instance, **MADDPG** in **Predator-Prey** and **MAPPO** in **Overcooked-AI** achieved high consistency scores of 0.81 and 0.89, respectively, indicating that agents closely followed the inferred roles. Robustness also improved, with **MAPPO** in **Overcooked-AI** achieving a robustness score of 0.89, up from 0.71 in the RB, underscoring the framework's impact on agents' resilience to perturbations.

However, one can point out a potential bias: organizational specifications were specifically designed to encompass all observations, avoiding non-handled new situations.

6.4 Comparison between MOISE+MARL and AGR+MARL

Table 2: Performance comparison between MOISE+MARL and AGR+MARL.

Framework	Env.	Conv. Rate	Robustness Score	Org. Fit	Cumulative Reward
MOISE+MARL	PP	0.85	0.83	0.87	245.8
AGR+MARL	PP	0.75	0.69	0.56	208.4
MOISE+MARL	OA	0.92	0.89	0.91	391.2
AGR+MARL	OA	0.82	0.75	0.58	348.9
MOISE+MARL	WM	0.88	0.86	0.90	307.1
AGR+MARL	WM	0.76	0.72	0.61	278.6

Table 2 highlights the impact of intermediary goals within MOISE+MARL. In **Overcooked-AI**, **MAPPO** under MOISE+MARL achieved a cumulative reward of 391.2, with an organizational fit of 0.91—33% higher than AGR+MARL's 0.58. Similarly, in **Warehouse Management**, **Q-Mix** under MOISE+MARL attained a cumulative reward of 307.1, an increase of nearly 10% over AGR+MARL's 278.6, with a higher robustness score (0.86 vs. 0.72).

Overall, these results underscore the importance of intermediary goals in fostering more stable, goal-oriented behaviors. By facilitating a clearer path to the global goal, MOISE+MARL consistently outperforms AGR+MARL in achieving higher rewards, robustness, and organizational fit across Predator-Prey (PP), Warehouse Management (WM), and Overcooked-AI (OA). Finally, we analyzed the impact of increasing the number of organizational constraints on training time. Preliminary results suggest a nearly linear growth in training duration as the number of constraints increases [1].

7 Conclusion and future works

The MOISE+MARL framework introduced in this paper aims to enhance control and explainability in MARL by incorporating organizational models that define explicit roles and missions for agents. Experimental results across several environments indicate that this framework helps agents adhere to expected behaviors while facilitating better policy convergence by constraining the policy search space. The results also show that agents trained with roles and goals exhibit behaviors closely resembling those determined via the framework, suggesting coherence between the application of organizational specifications and their expected effects.

However, the framework's reliance on predefined organizational specifications means it may struggle to adapt in highly dynamic or unstructured environments where agent roles and missions are less defined or evolve over time. Moreover, the computational overhead associated with enforcing organizational constraints and dynamically modifying rewards and actions may pose scalability challenges. Additionally, TEMM can be computationally intensive, which may hinder its applicability in real-time scenarios.

We are currently pursuing three main directions:

- Developing adaptive mechanisms that allow roles and missions to evolve dynamically during training, enabling agents to respond to changes in real-time
- Exploring automated methods, such as Large Language Models, for generating organizational specifications based on observed agent behaviors to help users on defining these specifications manually
- Improving the computational efficiency of TEMM or exploring alternative evaluation methods for real-world applications with larger agent populations.

Acknowledgments

This work was supported by *Thales Land Air Systems* within the framework of the *Cyb'Air* chair and the *AICA IWG*

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning*. 22–31.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. arXiv:1907.10902 [cs.LG] https://arxiv.org/abs/1907.10902

[3] Stefano V. Albrecht and Jacob Y. Foerster. 2024. Survey on Recent Advances in Cooperative Multi-Agent Reinforcement Learning. *Journal of Artificial Intelligence Research* (2024). to appear.

[4] Mohammed Alshiekh, Roderick Bloem, Matthew Johnson, James Kapinski, Keith Julian, and Mykel J Kochenderfer. 2018. Safe reinforcement learning via shielding. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (2018).

[5] Hamid R Berenji and David Vengerov. 2000. Learning, cooperation, and coordination in multi-agent systems. *Inference Systems Corporation, Technical report* (2000).

[6] Aurélie Beynier and Alain Mouaddib. 2013. A Decentralized Approach for Reinforcement Learning in Cooperative Multi-agent Systems. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 163–168.

[7] Micah Carroll, Rohin Shah, Mark Ho, Tom Griffiths, Pieter Abbeel, and Anca Dragan. 2020. Overcooked-AI: A Benchmark for Multi-Agent Learning under Partial Observability. *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), 2374–2380.

[8] Jacques Ferber, Olivier Gutknecht, and Fabien Michel. 2003. Agent/Group/Roles: Simulating with Organizations. In *ABS 2003 - 4th International Workshop on Agent-Based Simulation*, J.P. Muller (Ed.). Montpellier, France. https://hal-lirmm.ccsd.cnrs.fr/lirmm-00269714

[9] Jakob Foerster et al. 2018. Counterfactual multi-agent policy gradients. *International Conference on Machine Learning (ICML)* (2018).

[10] Jakob Foerster, Yannis Assael, Nando de Freitas, and Shimon Whiteson. 2018. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. *Advances in Neural Information Processing Systems* 31 (2018), 2137–2145.

[11] Javier Garcia and Fernando Fernandez. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16, 1 (2015), 1437–1480.

[12] Mohammad Ghavamzadeh and Sridhar Mahadevan. 2006. Hierarchical reinforcement learning with cooperative agents. In *Proceedings of the 23rd International Conference on Machine Learning*. 119–126.

[13] Qi Hu, Jun Chen, Jiajun Zhao, Zhenyu Xu, Xiaolin Liu, et al. 2021. MarlLib: A comprehensive library for multi-agent reinforcement learning. *arXiv preprint arXiv:2106.05912* (2021).

[14] Hubner, Jomi F et. al. 2007. Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *Int. Journal of Agent-Oriented Software Engineering* (2007), 370. https://doi.org/10.1504/ijaose.2007.016266

[15] A. Isakov, D. Peregorodiev, P. Brunko, and I. Tomilov. 2024. Cooperative-Competitive Decision-Making in Resource Management: A Reinforcement Learning Perspective. In *Advances in Machine Learning and Automated Learning*. Springer. https://doi.org/10.1007/978-3-031-77731-8_34

[16] Ariel Kwiatkowski, Mark Towers, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments. arXiv:2407.17032 [cs.LG] https://arxiv.org/abs/2407.17032

[17] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Advances in Neural Information Processing Systems* 30 (2017).

[18] K. Matsuyama, K. Su, J. Wang, D. Ye, and Z. Lu. 2025. CORD: Generalizable Cooperation via Role Diversity. *arXiv preprint* (2025). arXiv:2501.02221 https://arxiv.org/abs/2501.02221

[19] Frans A. Oliehoek and Christopher Amato. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer. https://link.springer.com/book/10.1007/978-3-319-28929-8

[20] Y. Qi, J. Cao, and B. Wu. 2024. Bidirectional Q-learning for recycling path planning of used appliances under strong and weak constraints. *Communications in Transportation Research* (2024). https://www.sciencedirect.com/science/article/pii/S2772424724000362

[21] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. *Proceedings of the 35th International Conference on Machine Learning* (2018), 4295–4304.

[22] Alex Ray, Joshua Achiam, and Dario Amodei. 2019. Benchmarking Safe Exploration in Deep Reinforcement Learning. In *arXiv preprint arXiv:1910.01708*.

[23] N. Sao Mai. 2024. *The intrinsic motivation of reinforcement and imitation learning for sequential tasks*. Ph.D. Dissertation. HAL Archive. https://hal.science/tel-04853270

[24] Jack Serrino, Max Kleiman-Weiner, et al. 2019. Finding Friend and Foe in Multi-Agent Games. In *Advances in Neural Information Processing Systems*.

[25] Maxwell Standen, Martin Lucas, David Bowman, Toby J. Richer, Junae Kim, and Damian Marriott. 2021. CybORG: A Gym for the Development of Autonomous Cyber Agents. arXiv:2108.09118 [cs.CR]

[26] Justin K Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Eugene Sulivan, Ruben Glatt Perez, Lukas Santos, Cameron Horsch, Christian Dieffendahl, et al. 2020. PettingZoo: Gym for multi-agent reinforcement learning. *Proceedings of the NeurIPS 2020 Track on Datasets and Benchmarks* (2020), 21–23.

[27] W. Wen, W. Long, P. Zhai, and L. Zhang. 2024. Role Play: Learning Adaptive Role-Specific Strategies in Multi-Agent Interactions. *arXiv preprint* (2024). arXiv:2411.01166 https://arxiv.org/abs/2411.01166

[28] Andrew Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. 2008. Learning and transferring roles in multi-agent MDPs. In *Proceedings of AAAI*. AAAI.

[29] Z. Xie, S. Shen, Y. Wang, C. Qiao, and B. Tang. 2024. Roco: Role-Oriented Communication for Efficient Multi-Agent Reinforcement Learning. *SSRN Electronic Journal* (2024). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5060074

[30] Chengjie Yu, Hao Dong, Yiqun Zhao, and Shuxin Zheng. 2021. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Advances in Neural Information Processing Systems* 34 (2021), 1091–1104.

[31] Sagir M Yusuf and Christopher Baber. 2020. Inferential Reasoning for Heterogeneous Multi-Agent Missions. *International Journal of Electrical and Computer Engineering* (2020).