Design and Evaluation of Multi-Agent AI System for Autonomous Decision Making

Satyam Kumar[1], Sagar Choudhary[2], Swati Jaiswal[3]

[*1,3]B.Tech Student, Department of CSE AI/ML, Quantum University, Roorkee, India.

[2]Assistant Professor, Department of CSE, Quantum University, Roorkee, India.

How to Cite this Article:
Kumar, S. & Jaiswal, S. (2026). Design and Evaluation of Multi-Agent AI System for Autonomous Decision Making. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.749

https://doi.org/ 10.55041/ijcope.v2i5.749

Abstract

Due to the rapid advancement in artificial intelligence technology, intelligent decision-making technology has gradually outperformed humans in many human versus machine contests, particularly in complex multi-agent collaborative task environments. In multi-agent collaboration decision-making, several agents cooperate to accomplish pre-defined tasks and realize certain goals. This technology can be applied in real-life scenarios like autonomous vehicles, drone navigation, disaster relief operations, and military confrontations simulations. This paper starts with a detailed review of the major simulation environments and platforms for multi-agent collaboration decision-making. We make a detailed analysis on the following aspects of these simulation environments: task format, reward distribution, and the technological base. Finally, we make an overall review of the intelligent decision-making methods and algorithms for multi-agent systems (MAS). They can generally be divided into five categories: rule-based (mainly fuzzy logic), game theory-based, evolutionary algorithm-based, deep MARL-based, and LLMs reasoning-based approaches. Considering that the MARL and LLMs-based decision-making approaches have a considerable edge over the conventional approaches like rule, game theory, and evolutionary algorithms, this paper aims to explore the multi-agent approaches based on MARL and LLMs. We offer a comprehensive review of such approaches, along with their methodologies, pros, and cons. Moreover, some future research directions related to multi-agent cooperative decision-making are also discussed.

Keywords:

Intelligent decision-making, Multi-agent systems, Multi-agent cooperative environments, Multi-agent reinforcement learning, Large language models.

1. Introduction

1.1. Research Backgrounds of Multi-Agent Decision-Making

With the continuous advancement of science and technology, intelligent decision-making technology has made rapid progress. These technologies have gradually surpassed human capabilities in various human machine game competitions, even exceeding the top human levels. Over the past few decades, especially following the successful application of Deep Q-Networks (DQN) [1] in the Arita game and the victories [2] of AlphaGo and Alpha Zero [3] over top human opponents[4], these landmark achievements have significantly propelled the advancement of intelligent decision-making research.

To meet the growing complexity of real-world applications and the increasing demand for more sophisticated, reliable, and efficient intelligent systems,

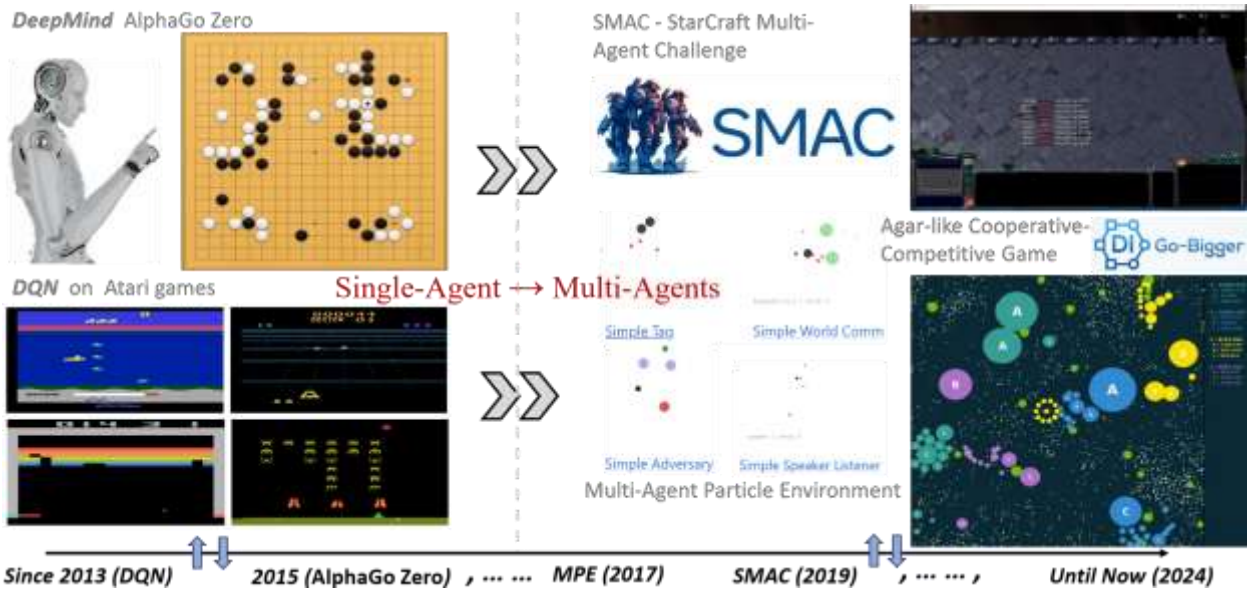

Figure 1: An overview of the evolution of scenarios and methods in decision-making from single-agent to multi-agent system.

Multiagent Cooperative Decision-Making Agent cooperative decision-making has developed very quickly into multiagent decision-making [5, 6, 7, 8]. Multiagent cooperative decision-making is an important area of machine learning (ML) [9] and artificial intelligence (AI) [10]. It deals with several agents collaborating to perform given tasks in dynamic simulated environments and in complex real-world systems.

Figure 1 illustrates the development progress in the area of multi-agent decision making from the point of view of a comparison between the current and past research methods, indicating that this highly dynamic research domain is an essential milestone on the way to human-like artificial intelligence (AI) and the Artificial General Intelligence (AGI) era. The applications of multi-agent cooperative decision making have numerous real-world usages and theoretical foundations. The potential areas are quite diverse, including intelligent agriculture management [11, 12], intelligent collaborative robots [13, 14, 15, 16], intelligent collaborative obstacle avoidance in self-driving cars [17, 18, 19], autonomous navigation [20, 21, 22], and joint rescue missions [12, 23]. Consequently, taking into account the fast pace of development and diverse requirements of the real world, this paper concentrates on the comprehensive investigation of multi-agent cooperative decision making.

1.2. Overview of Previous Multi-Agent Surveys

In line with rapid developments in the field of multi-agent cooperation for decision-making, there has been an observable trend toward conducting systematic literature reviews in this field [24, 6, 8, 25]. This includes a review of various aspects, from theory to application, offering a broad view of the current developments.

A systematic review by Ning et al. [25] covered the evolution, limitations, and applications of intelligent agents based on multiagent reinforcement learning (MARL). A literature review by Gronauer et al. [6] covered recent trends in multi-agent deep reinforcement learning, covering topics such as learning schemes, emergence of agent behavior, challenges associated with multi-agent domains, and future research directions. A literature survey by Yang et al. [26] considered the use of utility theory in AI robotics, concentrating on the role of utility-based AI systems in making decisions and promoting cooperation between multi-agent/robot systems. Recent progress in MARL, especially its applications in multi-robot systems, was covered by Orr et al. [8], who also discussed present challenges and future applications of MARL. A systematic review by Du et al. [24] focused on the application of multiagent deep reinforcement learning to MAS, highlighting its challenges, methods, and applications. The use of MARL in CAVs was comprehensively analyzed by Pamul et al. [7], who highlighted present developments, research directions, and challenges in this field.

1.3. Motivations of the Current Survey

Despite the increased interest in this research area, many surveys performed till now suffer from certain shortcomings [24, 6, 25, 28]. In particular, it can be said that our detailed analysis shows that the majority of existing reviews and surveys suffer from certain weaknesses:

- Research Scope Limitations: The previous literature review [27, 28] primarily stays confined to the scope of reinforcement learning theory and has not been able to overcome the boundaries of theory, thereby limiting its comprehensiveness.
- Ignoring Environments: Existing literature reviews [29, 6, 30] have mostly focused on advances in methodology and algorithms, often neglecting the important contribution made by simulation environments and platforms to multi-agent intelligent decision making.
- Undervaluing the Implementation Process: Previous studies [25, 28, 30] have emphasized theoretical frameworks without considering their implementation aspects, such as the code base and project architecture. Such an omission makes it difficult for readers to comprehend the implications of the findings. Undervaluing the Implementation Process: Previous studies [25, 28, 30] have emphasized theoretical frameworks without considering their implementation aspects, such as the code base and project architecture. Such an omission makes it difficult for readers to comprehend the implications of the findings.

1.4. The Survey Overview / Contents Organization

As shown in Figure 2, our survey has been designed based on our research methodology with each branch and sub-branch representing a specific section. First, in Section 1, we introduce the research background of multi agent cooperative decision-making and explain the limitations of past studies along with the organization of our current survey. Since the multi-agent reinforcement learning and language models-based intelligent decision-making approaches have distinct strengths and future possibilities, our focal areas lie on Deep Multi-Agent Reinforcement Learning-Based and Language Models-Based.

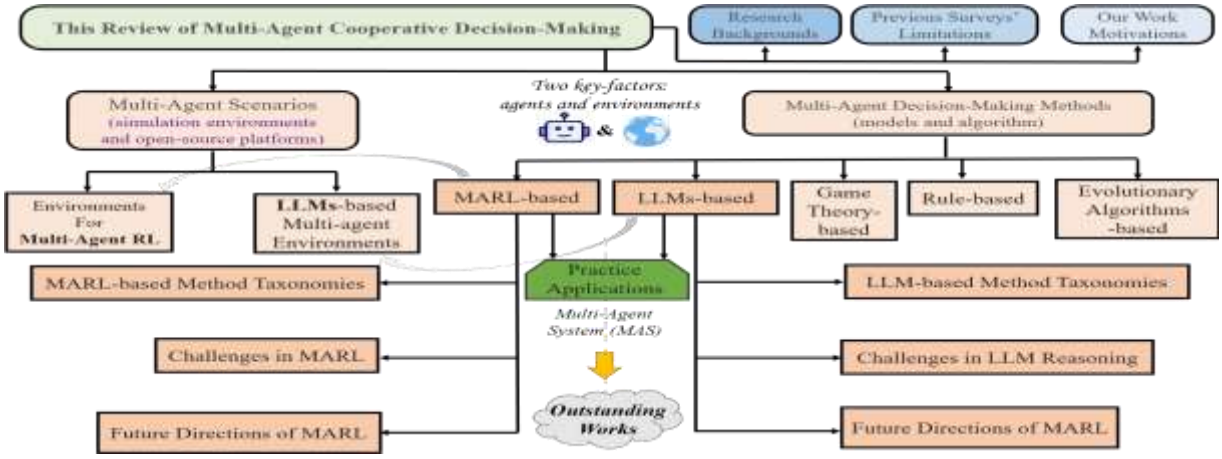

Figure 2: Illustration of our systematic review of multi-agent intelligent decision-making research. Compared to previous reviews, we have incorporated comprehensive introduction and analysis, with each segment corresponding to a specific chapter in the survey.

Methods will be preferred due to their higher capabilities of dealing with dynamic and unpredictable environments. Section 2 explores the popular intelligent decision-making methods, algorithms, and models. The methods will be categorized with emphasis on MARL-based and LLMs-based methods, detailing their methodologies, strengths, and weaknesses. Afterward, in Section 3, the major simulation environments for multi-agent cooperative decision-making will be analyzed.

2. Multi-Agent Decision-Making Taxonomies

In this part, some taxonomies of decision making for the multi-agent system as well as the techniques employed in each category will be elaborated. Generally, the multi-agent cooperative decision making techniques can be divided into five kinds: the rule-based technique (mostly based on fuzzy logic), game theory-based technique, the evolutionary algorithms-based method, MARL-based technique and LLMs-based technique [31]. Despite the effectiveness shown by these rule-based, game theory-based and evolutionary algorithms-based approaches, they all require the heavy use of pre-designed strategies and hypothesis.

Multi-perspective approach will be followed when the research is conducted in order to give a basis for further designing and optimisation of decision-making within MAS by applying the theories of agent interaction dynamics, main paradigms of cooperative decision-making, MARL (multi-agent reinforcement learning), and LLM (large language model)-based multi-agent systems.

2.1. Agent Interaction Dynamics for Multi-Agent Systems

Within multi-agent systems, the kind of interaction between different agents may be defined as relational dynamics of interactions, also referred to as agent interaction dynamics, which affect the behaviour of the whole system. This is crucial for intelligent systems whereby different agents interact in common settings. The different types of relations among agents include:

1. *Fully Cooperative:* In such an environment, all the agents will be having the same objectives. This means that they will have the same reward function. The agents will work in full coordination, trying to optimize their gains. Such environments are characterized by conditions where synergy is key. The performance of each individual agent is important for the success of other agents.

2. Fully Competitive: In this situation, the agents' relationship is described through the dynamics of zero-sum games, whereby the gains made by one party automatically mean losses for the other party. The agents are in direct confrontation due to the complete difference between their goals. This can be seen in situations such as robot races, whereby the robots are created to outdo others in performance.

3. *Mixed Cooperative and Competitive:* However, in most cases in the real world, agents are capable of exhibiting both forms of interactions, which is cooperation and competition. These are seen when agents are working in a group setting, such as robotic soccer; in such a case, agents are cooperating with other members in their own group towards a common goal (such as scoring) while competing against agents of other groups.

4. *Self-Interested:* For selfish behaviour dynamics, an agent will pursue its goals by attempting to maximize the utility associated with the goals without paying much attention to other agents or the effects that it causes to them. The agent can help or hinder another agent in the process, but that will not matter because the main focus will be its utility maximization. This will apply in a situation whereby agents are designed to be selfish such that maximizing their gains will not be optimal for the whole system.

2.2. Mainstream Paradigms of Multi-Agent Cooperative Decision-Making

There are many popular paradigms that can be used for solving problems in cooperative multi-agent decision-making by employing various methods. They use different techniques, such as rule-based systems (fuzzy logic) [32, 33, 34, 35, 36, 37], game-theory based [38, 39, 40, 41, 42, 43], evolutionary algorithms-based [44, 45, 46, 47, 48, 49], MARL-based [50], and language model-based [5730] multi-agent decision-making techniques. It should be noted that each of them has unique capabilities of its own and can be employed in specific scenarios and under particular conditions when dealing with autonomous agents.

2.2.1. Rule-Based (Primarily Fuzzy Logic)

Fuzzy logic is one rule-based system that has gained immense popularity within the field of MAS because it can accommodate ambiguity, inaccuracy of information, and unpredictable environment. Fuzzy logic allows the agent to make decisions as humans would do by associating linguistic rules with input variables.

A rule-based multi-agent control algorithm that uses local data rather than global coordinates was suggested by Miki et al. [32]. The rule-based multi-agent system to solve the coordination problems such as controllability and observability issues of distributed testing environments is suggested by Charaf et al. Yarahmadi et al. [33] surveyed the application of multiagent systems in CPS and IoT. They proposed a new approach for MAS that includes machine learning and rule-based reasoning approaches to enhance decision-making in MAS. The expert rule-based system that uses multi-agent technology for solving traffic management issues during weather problems is suggested by Marti et al.. Fuzzy logic along with Q-learning and game theory were used by Daeichian et al. to control traffic light in an autonomous manner. The fuzzy-theoretic game was developed by Wu et al. [34] that combines fuzzy logic with game theory to deal with uncertainty in utility values during MAS decision making. The cybersecurity management strategy for agricultural enterprises was designed by Nekhai et al. through developing multi-agent system (MAS) based on fuzzy logic reasoning.

Overall, fuzzy logic remains a foundational approach for rule-based decision-making in MAS, offering interpretability and robustness in uncertain environments. In the future, fuzzy logic will be further integrated with LLMs, hierarchical decision architectures, and multiagent planning, enabling more precise and adaptive decision-making in complex real-world scenarios.

2.2.2. ***Game theory-based***

The game theory offers an organized methodological approach for studying interactions among multiple intelligent agents. Agents can rationally solve problems based on the equilibrium optimization model [5, 29] in cooperation, competition, or both cases. The classical approaches include the Nash Equilibrium and the Stackelberg Game, and contemporary models employ concepts from reinforcement learning and Bayesian methods.

Game theoretic techniques for multiple agents were discussed by Wang et al. [38] with respect to both cooperative and non-cooperative approaches. Game theory was also utilized in multi-agent navigation and obstacle avoidance using Nash equilibrium by Guo et al. [39]. In addition, Zhang et al. formulated a decentralized controller for optimal coverage and network connectivity.

2.2.3. Evolutionary Algorithms-based

The use of evolutionary algorithms (EA) is a bio-mimicking method to optimize the functioning of multi-agent systems by taking advantage of biological concepts such as natural selection, mutation, and recombination [47]. The evolution of the agents' actions is a highly suitable strategy for cases involving constant learning, coordination, and self-organization.

The Multi-Agent Genetic Algorithm (MAGA) was developed by Liu et al. [45] for optimizing the global solution through competition and cooperation between the agents. MAGA was further modified into a hardware-based model by Xu et al. in which nanoclusters were used as agents for conducting massive parallel computations. Daan et al. [46] focused on the impact of evolutionary strategies in dynamic systems like finance, smart grids, and robotics.

2.2.4. MARL-based Multi-Agent Systems

Prior to the discussion of the MAS based on the MARL technique, a thorough discussion about the technological and methodological aspects of the single-agent systems that are based on the DRL technique, along with the MAS that use the MARL technique, will be discussed in detail in Appendix.

MARL provides a well-defined structure for the problem of decision-making in MAS, wherein autonomous agents are required to coordinate their actions among themselves as well as with respect to the environment. There are three main categories within which the MAS-related research in MARL can be divided. These include CTCE (Centralized Training with Centralized Execution DTDE (Decentralized Training with Decentralized Execution) and CTDE (Centralized Training with Decentralized Execution).

2.2.5. LLM-based Multi-Agent Systems

1) Despite their capacity for very large context lengths, LLMs such as GPT, Llama, and Gemini may vary widely in understanding complex inputs. On this note, the concept of collaboration among multi-agents is vital in improving performance since agents execute their operations based on task assignment. The agents work independently and may request assistance from other agents when required. In this regard, LLMs-based Multi-Agent Systems offer a fairly recent multi-agent decision-making framework which capitalizes on the power of language models to improve communication and collaboration between agents. An LLMs-based multi-agent system involves agents communicating in natural language or symbolic languages to decompose complex tasks into simpler subtasks. It is noteworthy that one crucial property of LLMs-based multi-agent systems is that it operates under a hierarchical arrangement of agents involving two levels:

2) Higher-level planners that take care of planning functions such as decomposition of tasks, allocation of resources, and strategy management.

3) Lower-level agents that execute particular subtasks and give feedback to the higher-level planner. They are mainly concerned about their subtasks, but still keep communication channels open with the higher levels to discuss their problems and make changes accordingly. This division makes distributed problem-solving feasible, where agents coordinate with each other through language, thereby executing their tasks collectively.

On the other hand, LLMs-based multi-agent systems have vast applications and great potential [30]. The collaboration among the robotic agents ensures their ability to do complex things like assembling products and conducting research. In autonomous driving, LLMs enable cars to interact, exchanging information regarding navigation strategies to coordinate their activities. Furthermore, agents such as drones can use LLMs to relay essential information to help the systems respond to emergencies. Wu et al. [103] developed AutoGen, an open-source tool for creating future LLM applications through conversational multi-agent systems, which customize the agent behavior and integrate LLMs, human interventions, and other components. Xiao et al. [101] designed Chain-of-Experts (CoE), a multi-agent system to reason about OR tasks through LLMs, assigning domain-specific roles and conductors to facilitate it. Chen et al. developed AgentVerse, a multiagent system motivated by group psychology in humans, modifying agent roles and configurations to solve complex tasks. Chen et al. designed AutoAgents, which automatically create and coordinate multiple specialists for effective performance on different tasks. Liu et alThe Dynamic LLM-Agent Network (DyLAN) [106] is an LLM-agent cooperation improvement method, which uses dynamic interaction depending on the task demands. CoELA (Cooperative Embodied Language Agent) is an improved way of cooperation among multiple agents for complicated, decentralized tasks, which utilizes the benefits of LLMs. MetaGPT [108, 109] is a meta-programming method for improvement of LLMs-based multi-agent systems by utilizing SOPs. XAgent [110] is an open-source project dedicated to development of autonomous agents for task solving by using LLMs and a dual-loop task planning and executing approach. PlanAgent is a closed-loop motion planning technique used for autonomous driving, where multiple LLMs provide hierarchical commands to the agent. LangGraph is a set of tools for developing stateful multi-actor applications by using LLMs, where workflow and state management are carefully controlled. CrewAI is an open-source framework for coordination of AI agents during their autonomous operation and role playing. Hou et al. CoAct was introduced by, which is a hierarchical multi-agent system that makes use of language models for collaborative task execution. The model comprises two agents, namely the global planning agent to plan tasks and formulate strategies for them, and the local execution agent to execute tasks and gather feedback.

Conclusion

To conclude, multi-agent systems based on LLMs have immense application possibilities in various fields and represent a sophisticated approach for resolving challenging decision-making tasks where agents need to coordinate and communicate with each other. Through enhancing cooperation processes between agents, LLMs increase the effectiveness of multi-agent systems significantly.

2.3. MARL-based Multi-Agent Decision-Making Taxonomies

However, in multi-agent systems with many autonomous agents interacting with the same environment and sometimes with one another, the decision process becomes far more complicated. In order for agents to optimize their actions, they have to not only learn how to act on their own but also how to cooperate with other agents. One of the main issues that arise in the context of MARL-based multi-agent system designs includes identifying an appropriate amount of information that needs to be exchanged between agents at various stages of training and deployment processes.

2.3.1. Centralized Training with Decentralized Execution (CTDE)

As depicted on the left-hand side of Figure 3, CTDE is a mixed MARL technique that utilizes the benefits of both centralized and decentralized techniques [124]. In CTDE, each agent has its own policy network, which is learned through the instructions of a centralized controller. CTDE is based on a two-step strategy: centralized learning and decentralized execution.

1. Algorithms Based on Value Decomposition The main problem that value decomposition-based algorithms solve in the area of multi-agent reinforcement learning is the estimation of the joint state-action value function (Q-value). Estimation of the joint value function becomes a difficult problem because of the large size of the action space. Instead of solving this problem, the value decomposition-based algorithms decompose the joint value function into individual state-action value functions (Qvalue), one for each agent. While operating, each agent performs an action, basing on its own value function. When the system is trained, the joint value function is estimated from the individual value functions, and the temporal difference error of the joint value estimates individual value functions. One of the conditions that must be satisfied by such algorithms is the Individual-Global-Max (IGM) condition.

One of the earliest value decomposition approaches to multi-agent reinforcement learning using cooperative task-decomposition framework is Value Decomposition Networks (VDN). In VDN, the complexity of estimating the joint state-action value function is reduced by assuming that the joint value function can be decomposed as a sum of individual state-action value functions for all the agents. This implies that estimating the joint value function becomes as simple as just taking the sum of all the individual value functions, without considering the different contributions of each individual agent's Q-function. This is because the assumption made by VDN constitutes a sufficient but unnecessary condition for satisfying the IGM property, thus limiting the usage of VDN. Furthermore, global state information is not employed by VDN in the training process.

2. Actor-Critic-based Approaches: Actor-Critic-based approaches stand out as a seminal set of techniques employed within the framework of CTDE, and provide an efficient toolset for overcoming the difficulties associated with multi-agent scenarios. Actor-Critic algorithms leverage the benefits of both policy gradient optimization (actor) and value function approximation (critic) in order to learn policies that adapt to a variety of cooperative and competitive tasks. Thanks to the utilization of a centralized critic in the learning process, Actor-Critic-based approaches help resolve crucial problems, including the ones related to environmental non-stationarity and credit assignment. The following is a brief discussion of some popular Actor-Critic approaches used in MARL.

MADDPG is a representative example of the application of Actor-Critic for CTDE tailored towards solving problems inherent to the multi-agent setting, where the interaction among the agents is collaborative and competitive at the same time. Traditional approaches like Q-learning and policy gradients fail in multi-agent scenarios because of problems such as non-stationarity where the environment keeps changing due to other agents' learning processes-and increased variance in the learning process as the number of agents grows. MADDPG makes an appropriate adjustment to the actor-critic approach by using a centralized critic during the training phase that has access to the actions and observation from all agents involved. Centralized critics help tackle the non-stationarity issue by allowing learning of a more stable value function that takes into account the entire action space. Nevertheless, when executing the policy, each agent operates independently using its policy function (or actor) based on observations of the environment made by the agent itself. It is possible to allow an agent to learn and execute sophisticated coordination behaviours with decentralized execution. In order to solve the computational issue of dealing with continuous action spaces, Li et al. have developed Multi-Agent Mutual Information Maximization Deep Deterministic Policy Gradient (M3DDPG).

3. PPO based Approaches: Proximal Policy Optimization (PPO) is one of the popular CTDE based reinforcement learning algorithms, which have been customized and generalized to solve problems encountered in the context of MARL. In the context of CTDE methods, it has been observed that PPO along with multi-agent extensions of PPO are highly effective in achieving policy improvement efficiency and stability. The PPO method has been introduced by Schulman et al. as an efficient policy gradient method intended to enhance the existing trust region policy optimization (TRPO) approach [159]. In PPO, a clipped surrogate objective function is used, which allows simplification of the constraints involved in TRPO while still ensuring stability of the method without imposing heavy computation cost.

In the context of MARL, the Multi-Agent Proximal Policy Optimization (MAPPO) algorithm is a version of the PPO algorithm applied to the centralized critic framework. The MAPPO algorithm utilizes a centralized value function (or critic) that assesses joint states and joint actions, whereas individual agents follow independent decentralized policies. The MAPPO approach shows better results in different types of cooperative and competitive multi-agent scenarios, including, but not limited to, StarCraft II environment and MultiAgent Particle Environment (MPE). Nevertheless, although MAPPO uses parameter sharing, this assumption can be irrelevant to a heterogeneous group of agents.

4. Other Kinds of Algorithms used in the CTDE Approach: Apart from the traditional algorithms categorized into Value Decomposition algorithms, Actor-Critic algorithms, and PPO-based algorithms, other kinds of algorithms have been developed in the MARL research, which have made considerable improvements through novel optimizations and enhancements within the framework of CTDE, but which cannot be classified in any of these traditional kinds. This is because these algorithms attempt to overcome the limitations of multi-agent settings, including non-stationarity and poor communication.

Another example is Centralized Advising and Decentralized Pruning (CADP), which is a new paradigm presented by Zhou et al. due to the weaknesses in the CTDE paradigm. The framework improves the training process by enabling agents to share and advise each other during centralized training, making joint-policy search possible. To ensure decentralized execution, the framework uses a smooth pruning model, which restricts the interaction between agents while preserving cooperation abilities and showcasing better results than other models in multiagent StarCraft II SMAC and Google Research Football games. CommNet offers a neural network approach where multiple agents are capable of communicating simultaneously through an established channel, thereby maximizing their efficiency in fully cooperative tasks. In this manner, agents can come up with their own communication methods during the training process. Another approach in game theoretic Meta-Learning for MARL was the Meta-MARL framework by Mao et al.

2.3.2. Decentralized Training with Decentralized Execution (DTDE)

DTDE is illustrated at the Center of Fig. 3. It is a purely decentralized algorithm where agents act individually in their environment and update their policies according to the individual experiences and rewards gained by each individual agent. As can be seen, in DTDE, every agent trains and acts independently, using only its own experiences and rewards to modify its strategy. The algorithm is especially useful when there is restricted communication and no global coordination in the environment.

- Distributed Q-Learning optimistically assumes other agents always take optimal actions, focusing on learning from high-reward interactions. While effective in deterministic settings, it can be overly optimistic in environments with randomness.

- Hysteretic Q-Learning By introducing two learning rates—one for positive updates and another, smaller rate for negative updates—hysteretic Q-learning balances optimism with robustness in stochastic environments.

- Lenient Q-Learning dynamically adjusts how lenient the agent is in updating its values, depending on how frequently specific state-action pairs are encountered. It allows for more exploration in the early stages of learning while focusing on optimization later.

With increasing complexity in MARL problems, approaches based on DTDE techniques have been extended to deep learning. The most successful approaches involve applying Deep Q-Networks (DQN) and Deep Recurrent Q-Networks (DRQN) in decentralized settings to allow agents to operate in high dimensional spaces. One such technique is Independent DRQN (IDRQN), which utilizes an asynchronous technique for experience replay, leading to the potential problem of instability. This issue is addressed using the use of CERTs that synchronize experience replay processes among agents, decreasing non-stationarity. Dec-HDRQN is a hysteretic version of DRQN that utilizes deep neural network techniques along with concurrent buffers in a decentralized manner to address problems in a partially observable environment.

In the DTDE paradigm, policy gradient methods offer an alternative to value-based approaches, particularly for scenarios involving continuous action spaces. Several policy gradient DTDE methods have been proposed:

- Decentralized REINFORCE independently optimizes each agent's policy using gradient ascent based on rewards observed during episodes. While simple, it is less sample-efficient.

- Independent Actor-Critic (IAC) Combining value estimation (critic) and policy optimization (actor), IAC enables agents to learn faster and update more frequently, improving sample efficiency.

- Independent Proximal Policy Optimization (IPPO) Extending Proximal Policy Optimization (PPO) to decentralized settings, IPPO improves policy stability by limiting how much policies can change between updates.

Despite its advantages, DTDE still faces significant challenges: 1. *Non-Stationarity*: As other agents learn and adapt, the environment appears dynamic and unstable to each agent, making convergence difficult; 2. *Credit Assignment*: It is hard to determine how each agent's actions contribute to the team's overall reward in cooperative tasks; 3. *Trade-Offs Between Scalability and Performance*: While DTDE scales well, its performance may be limited in tasks requiring high levels of coordination. To overcome these challenges, future research could focus on improving communication strategies during training and more robust strategies for dynamic environments.

Conclusion

Overall, the DTDE approach can be seen as an effective solution for dealing with distributed decision-making issues while achieving a proper balance between scalability, independence, and efficiency. The concept has already proved its effectiveness in applications like autonomous driving, energy distribution networks, and robot swarms. Moving forward, it is reasonable to expect that DTDE will see more widespread application in practical multi-agent systems.

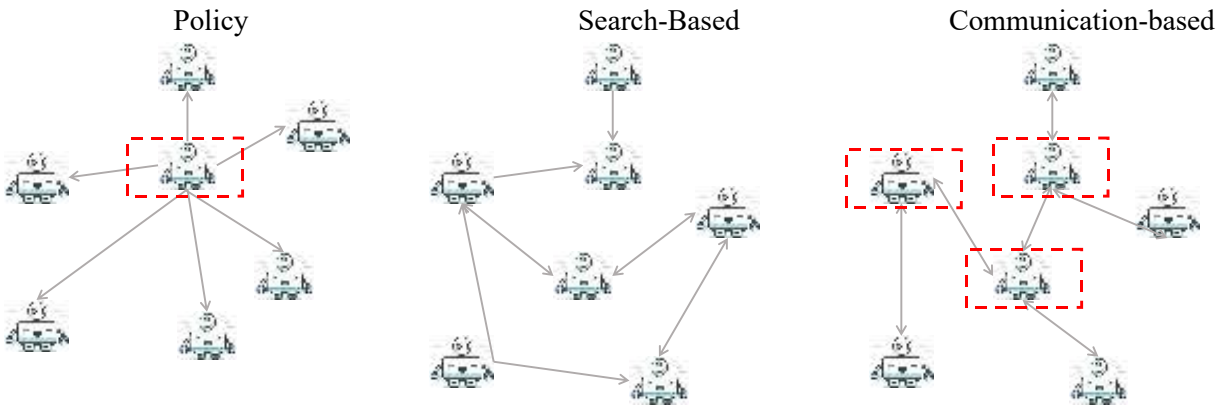

Figure 4: A schematic representation of three distinct communication methods among agents, with arrc indicating the direction of message transmission. (a) *Broadcasting communication*: The activated ag transmits messages to all other agents within the communication network. (b) *Targeted communicati* Agents selectively communicate with specific target agents based on a supervisory mechanism that regula the timing, content, and recipients of the messages. (c) *Networked communication*: Agents engage in locali interactions with their neighboring agents
within the network.

MARL: Concerning Communication-based MARL Algorithms that Use Policy Search,
several advancements have been made using algorithms like Communication Networks (CommNet), Bidirectional Coordinated Network (BiCNet),
Multi-Agent Distributed MADDPG (MD-MADDPG),
Intrinsic A3C, and Multi-Agent Communication and Coordination (MACC).
Some of these include CommNet, which provides a centralized yet
differentiable approach of communication whereby agents send signals to form global contexts.
On the other hand, BiCNet is a method whereby agents are allowed to communicate through
bidirectional recurrent layers; hence, it is best suited for complicated assignments. MD-MADDPG uses a technique known as centralized training and decentralized execution,
allowing the agents to exchange vital information during training. Intrinsic A3C involves the introduction of intrinsic motivation to facilitate effective exploration of tasks,
agents sharing intrinsic rewards in the process. Lastly, Multi-Agent Communication and Coordination (MACC) involves developing adaptive communication, thus providing stability and security during the coordination process.

MARL with Communication Improving Communication Efficiency: Among MARL algorithms targeting improving communication efficiency, some excellent strategies are Attentional Communication (ATOC) [138], Targeted Multi-Agent Communication (TarMAC), and Inter-Agent Centralized Communication (IC3Net) [178]. Attentional Communication (ATOC) utilizes an attention strategy to identify occasions where communication is required, thus achieving efficiency and coordination. Targeted Multi-Agent Communication (TarMAC) [139] applies targeted attention strategies that help agents send communications to teammates when they are relevant, reducing unnecessary communication and ensuring high performance. IC3Net involves a gating strategy to ensure the timing of communication by learning its necessity.

These research advances in Communication-based MARL methods demonstrate significant strides in enabling agents to share information and achieve coordinated decision-making in MAS. These advancements will pave the way for deploying MARL in real-world scenarios where efficient and effective communication is essential.

2.4. LLMs-based Multi-Agent System Taxonomies

The field of LLMs-based multi-agent systems has seen significant advancements, with researchers exploring various aspects of these systems to enhance their capabilities and applications [30]. A comprehensive taxonomy can help categorize and understand the different dimensions of LLMs-based multi-agent systems, including architectural design, application domains, evaluation methods, and future research directions.

2.4.1. Architectural Design

The design of architectures for LLMs-based multiagent systems is a critical component in harnessing the full potential of LLMs to enhance the capabilities of autonomous agents. Architectural design encompasses the framework and mechanisms that enable agents to interact, adapt, and make decisions in complex and dynamic environments. This section explores two primary levels of autonomy within these systems: Adaptive Autonomy and Self-Organizing Autonomy.

- Adaptive Autonomy: Adaptive autonomy refers to systems where agents can adjust their behavior within a predefined framework. These agents are designed to operate within the constraints set by the system architects but can adapt their actions based on the specific requirements of the task at hand. For example, in a taskspecific adaptation scenario, an agent might adjust its search strategy in an information retrieval task based on the relevance of the results. In a contextaware adaptation scenario, an agent might change its communication style based on the social context of the interaction. This level of autonomy is crucial for agents that need to operate in dynamic environments where the task requirements can change over time.

- Self-Organizing Autonomy: Self-organizing autonomy represents a higher level of autonomy where agents can dynamically adapt their behavior without predefined structures. This allows for more flexible and context-aware interactions among agents. For instance, in dynamic task allocation, agents can assign tasks to each other based on the current state of the environment and their individual skills. Emergent behavior is another key feature at this level, where agents can form coalitions or develop new strategies to solve complex problems. This level of autonomy is essential for multi-agent systems that need to operate in highly dynamic and unpredictable environments.

2.4.2. Applications

In the social sciences, LLMs-based agents have been used to simulate various social phenomena, providing insights into human behavior and social dynamics.

- *1) Economic Agents:* LLMs can be used to model economic agents, similar to how economists use the concept of homo economicus.

Experiments have shown that LLMs can produce results qualitatively similar to those of traditional economic models, making them a promising tool for exploring new social science insights. For example, in market simulation, LLMs can predict market trends and the impact of economic policies. In behavioural economics, LLMs can model individual and group decision-making processes, providing a more nuanced understanding of economic behaviour.

- *2) Social Network Simulation:* The Social-network Simulation System (S3) uses LLMs-based agents to simulate social networks, accurately replicating individual attitudes, emotions, and behaviors. This system can model the propagation of information, attitudes, and emotions at the population level, providing valuable insights into social dynamics. For example, it can simulate how information spreads through social networks and identify influential nodes, or model the evolution of social norms and behaviours over time.

- *3) User Behavior Analysis:* LLMs are employed for user simulation in recommender systems, demonstrating superiority over baseline simulation systems. They can generate reliable user behaviors, improving the accuracy of recommendations. For example, in personalized recommendations, LLMs can generate user profiles and behaviors to optimize recommendation algorithms. In user engagement, LLMs can simulate user interactions to optimize user retention and engagement.

3. *Simulation Environments of Multi-Agent Decision-Making*

First and foremost, the designs and implementations of multi-agent cooperative simulation environments are crucial in the historical research of multiagent decision-making, which are widely utilized in practical applications and production. These simulation environments form the foundation for conducting efficient and effective studies in multi-agent cooperative decision-making. Specifically, a dynamic multiagent cooperative decision-making environment refers to predetermined scenarios and platforms where multiple agents collaborate to solve problems, complete tasks, and achieve goals. Such environments provide not only a platform for testing and validating various intelligent decision-making algorithms but also help us better understand the behaviours and interactions of agents in dynamic settings. By simulating these interactions, researchers can gain insights into how agents coordinate and adapt to changing conditions, thereby improving the robustness and efficiency of multi-agent systems in real-world applications. Consequently, the importance of these simulation environments cannot be overstated. They serve as a testing ground for theoretical models, allowing researchers to observe the practical implications of their intelligent algorithms. Additionally, these platforms help in identifying potential issues and refining strategies before deployment in actual scenarios, ensuring that the agents are well-prepared to handle the complexities of real-world environments. In Table 2, a wide range of simulated environments is listed. Next, we will delve into these environments one by one, emphasizing their significance and features for future development.

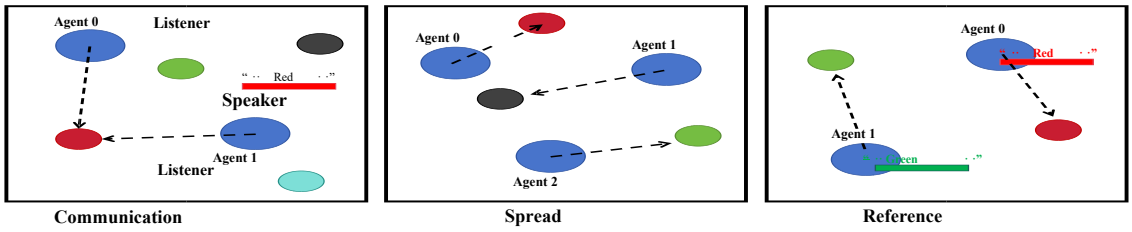

Figure 5: Typical Scenarios in Multi-Agent Particle Environment (MPE).

3.1. Several Widely-used Environments on MARL

Multi-Agent Particle Environment (MPE) is a versatile and widely-used MARL platform designed for research in both cooperative and competitive settings. Developed by OpenAI, it is primarily known for being the testing environment of the MADDPG algorithm. MPE is a time-discrete, space continuous 2D platform designed for evaluating MARL algorithms.

Overall, MPE is a pivotal resource in the MARL community, offering a well-rounded platform for experimentation and algorithm comparison. Its design and functionality have made it an indispensable tool for researchers seeking to push the boundaries of what is possible in multi-agent systems.

StarCraft Multi-Agent Challenge (SMAC)[1] is a widely-used benchmark for MARL that focuses on decentralized micromanagement tasks in the popular real-time strategy game StarCraft II[2]. In SMAC, multiple agents control

individual units and must learn to cooperate and coordinate actions based on local, partial observations. The agents face complex challenges, including coordinating combat techniques like focus fire, kiting, and positioning, while the opponent is controlled by the built-in StarCraft II AI. SMAC emphasizes problems such as partial.

Figure 6: Several Typical Scenarios in StarCraft Multi-Agent Challenge (SMAC).

StarCraft Multi-Agent Challenge 2 (SMACv2)[3] [134, 154, 91] However, SMAC has limitations, including insufficient stochasticity and partial observability, which allows agents to perform well with simple open-loop policies. To address these shortcomings, SMACv2 introduces *procedural content generation (PCG)*, randomizing team compositions and agent positions, ensuring agents face novel, diverse scenarios. Several multi-agent decision-making scenarios are depicted in Figure 7, which are from Benjamin et al. [135]. This requires more sophisticated, closed-loop policies that condition on both ally and enemy information. Additionally, SMACv2 includes the *Extended Partial Observability Challenge (EPO)*, where enemy observations are masked stochastically, forcing agents to adapt to incomplete information and communicate more effectively. SMACv2 thus represents a major evolution of the original benchmark, addressing key gaps such as the lack of stochasticity and meaningful partial observability. These changes make SMACv2 a more challenging environment, requiring agents to generalize across varied settings and improve coordination, communication, and decentralized decision-making. Overall, SMACv2 provides a more rigorous testbed for advancing the field of cooperative MARL.

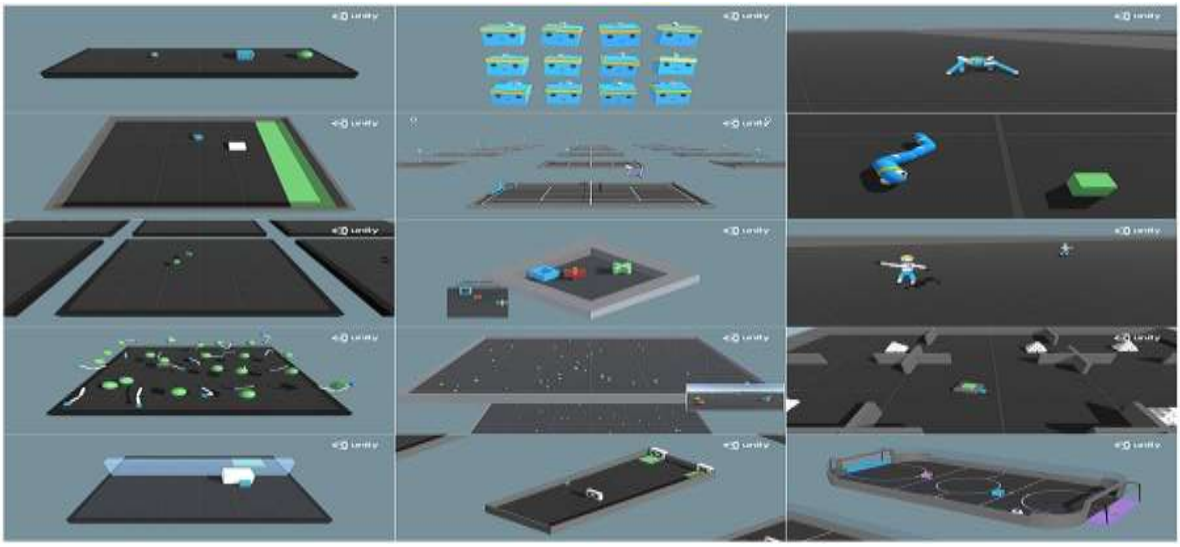

Figure 9: Typical Training Scenarios in Unity Machine LearningAgents Toolkit (released version: v0.11). From Left-to-right, up-todown: (a) Basic, (b) 3DBall, (c) Crawler, (d) Push Block, (e) Tennis, (f) Worm, (g) Bouncer, (h) Grid World, (i) Walker, (j) Reacher, (k) Food Collector, (l) Pyramids, (m) Wall Jump, (n) Hallway, (o) Soccer Twos [194].

ML-Agents is particularly useful for training NPC behaviors in diverse scenarios, automated testing of game builds, and evaluating game design decisions. It features a highly flexible simulation environment with realistic visuals, physics-driven interactions, and rich task complexity. By integrating tools for creating custom environments and supporting multi-agent and adversarial settings, the toolkit bridges the gap between AI research and practical applications in game development.
As seen from Figure 9, it depicts several typical multi-agent environments from the previous work of Juliani et al. The platform also provides key components such as a Python API, Unity SDK, and pre-built environments, enabling users to customize and evaluate their algorithms in Unity's interactive and visually rich settings. With its versatility and accessibility, Unity ML-Agents Toolkit has become an indispensable resource for both AI researchers and game developers, driving innovation in artificial intelligence and simulation-based learning.

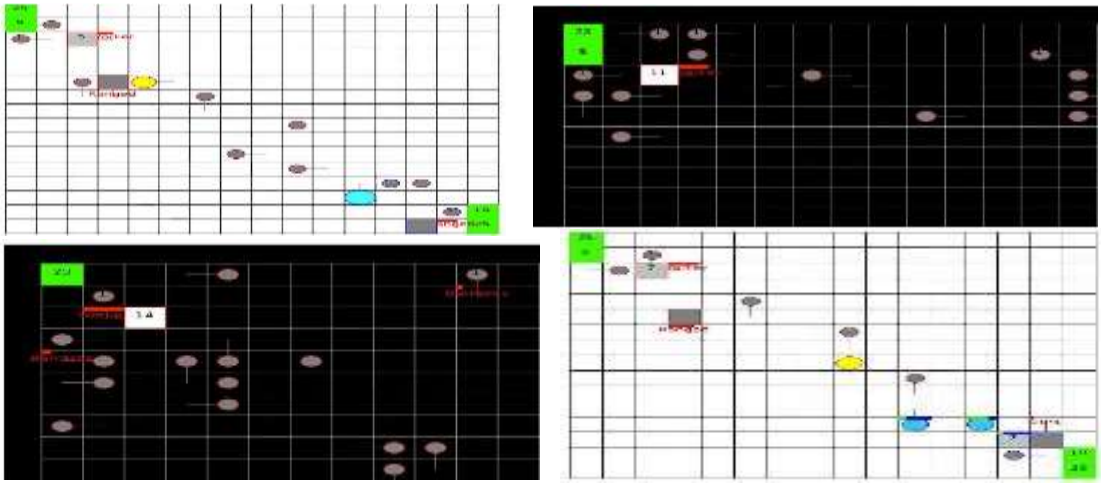

Figure 10: Screenshot of our best-trained agent (top-left) playing against CoacAI (bottom-right), the 2020 µRTS AI competition champion [183].

3.2. LLMs Reasoning-based Simulation Environments

LLMs-based multi-agent systems have become an essential tool for enhancing the collaboration, reasoning, and decision-making capabilities of autonomous agents. By integrating LLMs with simulation platforms, researchers can create complex test environments to explore the interactions of multi-agent systems in various tasks and scenarios. These simulation environments not only provide rich dynamic testing scenarios but also promote the widespread application of LLMs in task planning, coordination, and execution. The following will introduce several widely used simulation platforms for LLM multi-agent systems.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning (2013). arXiv:1312.5602.
URL https://arxiv.org/abs/1312.5602

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533. doi:10.1038/ nature14236.
URL https://doi.org/10.1038/nature14236

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489. doi:10.1038/nature16961.
URL https://doi.org/10.1038/nature16961

[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou,
A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, D. Hassabis, Mastering the game of go without human knowledge, Nature 550 (7676) (2017) 354–359. doi:10.1038/nature24270. URL https://doi.org/10.1038/nature24270

[5] T. Li, K. Zhu, N. C. Luong, D. Niyato, Q. Wu, Y. Zhang, B. Chen, Applications of multi-agent reinforcement learning in future internet: A comprehensive survey, IEEE Communications Surveys & Tutorials 24 (2) (2022) 1240–1279. doi:
10.1109/COMST.2022.3160697.

[6] S. Gronauer, K. Diepold, Multi-agent deep reinforcement learning: a survey, Artificial Intelligence Review 55 (2) (2022)
895–943. doi:10.1007/s10462-021-09996-w.
URL https://doi.org/10.1007/s10462-021-09996-w

[7] P. Yadav, A. Mishra, S. Kim, A comprehensive survey on multi-agent reinforcement learning for connected and automated vehicles, Sensors 23 (10) (2023). doi:10.3390/ s23104710.

[8] J. Orr, A. Dutta, Multi-agent deep reinforcement learning for multi-robot applications: A survey, Sensors 23 (7) (2023). doi:10.3390/s23073625.
URL https://www.mdpi.com/1424-8220/23/7/3625

[9] W. Jin, B. Zhao, Y. Zhang, J. Huang, H. Yu, Wordtransabsa: Enhancing aspect-based sentiment analysis with masked language modeling for affective token prediction, Expert Systems with Applications 238 (2024) 122289. doi:https://doi.org/10.1016/j.eswa.2023.122289. URL https://www.sciencedirect.com/science/ article/pii/S0957417423027914

[10] B. Zhao, W. Jin, Y. Zhang, S. Huang, G. Yang, Prompt learning for metonymy resolution: Enhancing performance with internal prior knowledge of pre-trained language models, Knowledge-Based Systems 279 (2023) 110928. doi:https:

//doi.org/10.1016/j.knosys.2023.110928. URL https://www.sciencedirect.com/science/ article/pii/S0950705123006780

[11] A. Seewald, C. J. Lerch, M. Chancan, A. M. Dollar, I. Abra-´ ham, Energy-aware ergodic search: Continuous exploration for multi-agent systems with battery constraints (2024). arXiv:
2310.09470.
URL https://arxiv.org/abs/2310.09470

[12] M. M. H. Qazzaz, S. A. R. Zaidi, D. C. McLernon, A. Salama, A. A. Al-Hameed, Optimizing search and rescue uav connectivity in challenging terrain through multi q-learning (2024).
arXiv:2405.10042.
URL https://arxiv.org/abs/2405.10042

[13] G. T. Papadopoulos, M. Antona, C. Stephanidis, Towards open and expandable cognitive ai architectures for large-scale multi-agent human-robot collaborative learning, IEEE Access 9 (2021) 73890–73909. doi:10.1109/ACCESS.2021.
3080517.

[14] M. D. Silva, R. Regnier, M. Makarov, G. Avrin, D. Dumur, Evaluation of intelligent collaborative robots: a review, in: 2023 IEEE/SICE International Symposium on System Integration (SII), 2023, pp. 1–7. doi:10.1109/SII55687.2023.
10039365.

[15] Y. Huang, S. Wu, Z. Mu, X. Long, S. Chu, G. Zhao, A multiagent reinforcement learning method for swarm robots in space collaborative exploration, in: 2020 6th International Conference on Control, Automation and Robotics (ICCAR), 2020, pp. 139–144. doi:10.1109/ICCAR49639.2020.9107997.

[16] C. G. Cena, P. F. Cardenas, R. S. Pazmino, L. Puglisi, R. A. Santonja, A cooperative multi-agent robotics system: Design and modelling, Expert Systems with Applications 40 (12) (2013) 4737–4748. doi:https:
//doi.org/10.1016/j.eswa.2013.01.048.
URL https://www.sciencedirect.com/science/ article/pii/S0957417413000791

[17] S. Jayawardana, V. Jayawardana, K. Vidanage, C. Wu, Multibehavior learning for socially compatible autonomous driving, in: 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), 2023, pp. 4422–4427. doi:10.1109/ITSC57777.2023.10422120.

[18] L. Wen, J. Duan, S. E. Li, S. Xu, H. Peng, Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020, pp.
1–7. doi:10.1109/ITSC45102.2020.9294262.

[19] L. Weiwei, H. Wenxuan, J. Wei, L. Lanxin, G. Lingping, L. Yong, Learning to model diverse driving behaviors in highly interactive autonomous driving scenarios with multi-agent reinforcement learning (2024). arXiv:2402.13481.
URL https://arxiv.org/abs/2402.13481

[20] Y. Xue, W. Chen, Multi-agent deep reinforcement learning for uavs navigation in unknown complex environment, IEEE Transactions on Intelligent Vehicles 9 (1) (2024) 2290–2303. doi:10.1109/TIV.2023.3298292.

[21] S. Rezwan, W. Choi, Artificial intelligence approaches for uav navigation: Recent advances and future challenges, IEEE Access 10 (2022) 26320–26339. doi:10.1109/ACCESS.2022.
3157626.

[22] B. Al Baroomi, T. Myo, M. R. Ahmed, A. Al Shibli, M. H. Marhaban, M. S. Kaiser, Ant colony optimization-based path planning for uav navigation in dynamic environments, in: 2023 7th International Conference on Automation, Control and Robots (ICACR), 2023, pp. 168–173. doi:10.1109/
ICACR59381.2023.10314603.

[23] T. Samad, S. Iqbal, A. W. Malik, O. Arif, P. Bloodsworth, A multi-agent framework for cloud-based management of collaborative robots, International Journal of Advanced Robotic Systems 15 (4) (2018). doi:10.1177/1729881418785073.

[24] W. Du, S. Ding, A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications, Artificial Intelligence Review 54 (5) (2021) 3215–3238. doi:
10.1007/s10462-020-09938-y.
URL https://doi.org/10.1007/s10462-020-09938-y

[25] Z. Ning, L. Xie, A survey on multi-agent reinforcement learning and its application, Journal of Automation and Intelligence 3 (2) (2024) 73–91. doi:https://doi.org/10.1016/j.jai.2024.02.003. URL https://www.sciencedirect.com/science/ article/pii/S2949855424000042

[26] Q. Yang, R. Liu, Understanding the application of utility theory in robotics and artificial intelligence: A survey (2023). arXiv:
2306.09445.
URL https://arxiv.org/abs/2306.09445

[27] P. Hernandez-Leal, M. Kaisers, T. Baarslag, E. M. de Cote, A survey of learning in multiagent environments: Dealing with non-stationarity (2019). arXiv:1707.09183.
URL https://arxiv.org/abs/1707.09183

[28] C. Zhu, M. Dastani, S. Wang, A survey of multi-agent deep reinforcement learning with communication, Autonomous Agents and Multi-Agent Systems 38 (1) (2024) 4. doi:
10.1007/s10458-023-09633-6.
URL https://doi.org/10.1007/s10458-023-09633-6

[29] T. T. Nguyen, N. D. Nguyen, S. Nahavandi, Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications, IEEE Transactions on Cybernetics 50 (9) (2020) 3826–3839. doi:10.1109/TCYB.2020.
2977374.

[30] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang,
Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, J. Wen, A survey on large language model based autonomous agents, Frontiers of Computer Science 18 (6) (2024) 186345. doi:10.1007/s11704-024-40231-1. URL https://doi.org/10.1007/s11704-024-40231-1

[31] B. Zhao, W. Jin, J. Del Ser, G. Yang, Chatagri: Exploring potentials of chatgpt on cross-linguistic agricultural text classification, Neurocomputing 557 (2023) 126708. doi:
https://doi.org/10.1016/j.neucom.2023.126708. URL https://www.sciencedirect.com/science/ article/pii/S0925231223008317

[32] T. Miki, M. Nagao, H. Kobayashi, T. Nakamura, A simple rule based multi-agent control algorithm and its implementation using autonomous mobile robots, in: 2010 World Automation Congress, 2010, pp. 1–6.

[33] H. Yarahmadi, H. Navidi, M. Challenger, Improving the resource allocation in iot systems based on the integration of reinforcement learning and rule-based approaches in multiagent systems, in: 2024 8th International Conference on Smart Cities, Internet of Things and Applications (SCIoT), 2024, pp.
135–141. doi:10.1109/SCIoT62588.2024.10570102.

[34] S.-H. Wu, V.-W. Soo, A fuzzy game theoretic approach to multi-agent coordination, in: T. Ishida (Ed.), Multiagent Platforms, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 76–87. doi:10.1007/3-540-48826-X_6.

[35] H. Zhang, J. Zhang, G.-H. Yang, Y. Luo, Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming, IEEE Transactions on Fuzzy Systems 23 (1) (2015) 152–163. doi:10.1109/TFUZZ.2014.2310238.

[36] F. Ren, M. Zhang, Q. Bai, A fuzzy-based approach for partner selection in multi-agent systems, in: 6th IEEE/ACIS Interna-

tional Conference on Computer and Information Science (ICIS 2007), 2007, pp. 457–462. doi:10.1109/ICIS.2007.21.

[37] D. Gu, H. Hu, Fuzzy multi-agent cooperative q-learning, in: 2005 IEEE International Conference on Information Acquisition, 2005, p. 5 pp. doi:10.1109/ICIA.2005.1635080.

[38] J. Wang, Y. Hong, J. Wang, J. Xu, Y. Tang, Q.-L. Han,

J. Kurths, Cooperative and competitive multi-agent systems: From optimization to games, IEEE/CAA Journal of Automatica Sinica 9 (5) (2022) 763–783. doi:10.1109/JAS.2022.

105506.

[39] Y. Guo, Q. Pan, Q. Sun, C. Zhao, D. Wang, M. Feng, Cooperative game-based multi-agent path planning with obstacle avoidance, in: 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), 2019, pp. 1385–1390. doi:

10.1109/ISIE.2019.8781205.

[40] D. Schwung, A. Schwung, S. X. Ding, Distributed selfoptimization of modular production units: A state-based potential game approach, IEEE Transactions on Cybernetics 52 (4) (2022) 2174–2185. doi:10.1109/TCYB.2020.3006620.

[41] X. Wang, J. Wang, J. Chen, Y. Yang, L. Kong, X. Liu, L. Jia, Y. Xu, A game-theoretic learning framework for multi-agent intelligent wireless networks (2019). arXiv:1812.01267.

URL https://arxiv.org/abs/1812.01267

[42] W. Lin, Y. Chen, Q. Q. Wang, J. Zeng, J. Liu, Multi-agents based distributed-energy-resource management for intelligent microgrid with potential game algorithm, in: IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, 2017, pp. 7795–7800. doi:10.1109/IECON.2017.

8217366.

[43] H. Wang, Z. Ning, H. Luo, Y. Jiang, M. Huo, Game-based adaptive optimization approach for multi-agent systems, in: 2023 IEEE International Conference on Industrial Technology (ICIT), 2023, pp. 1–5. doi:10.1109/ICIT58465.2023.

10143172.

[44] L. Bull, Evolutionary computing in multi-agent environments: Operators, in: V. W. Porto, N. Saravanan, D. Waagen, A. E. Eiben (Eds.), Evolutionary Programming VII, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 43–52.

[45] J. Liu, W. Zhong, L. Jiao, Multi-Agent Evolutionary Model for Global Numerical Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 13–48. doi:10.1007/

978-3-642-13425-8_2.

URL https://doi.org/10.1007/978-3-642-13425-8_2

[46] D. Bloembergen, K. Tuyls, D. Hennes, M. Kaisers, Evolutionary dynamics of multi-agent learning: a survey, J. Artif. Int. Res. 53 (1) (2015) 659–697.

[47] D. Klijn, A. E. Eiben, A coevolutionary approach to deep multi-agent reinforcement learning, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 283–284. doi:10.1145/3449726.

3459576.

URL https://doi.org/10.1145/3449726.3459576 [48] S. Yuan, K. Song, J. Chen, X. Tan, D. Li, D. Yang, Evoagent: Towards automatic multi-agent generation via evolutionary algorithms (2024). arXiv:2406.14228.

URL https://arxiv.org/abs/2406.14228

[49] W. Zhang, H. Liu, Evolutionary game analysis of multi-agent cooperation strategy analysis in agricultural water conservancy ppp project under digitization background, Scientific Reports 14 (1) (2024) 22915. doi:10.1038/s41598-024-74065-5.

URL https://doi.org/10.1038/s41598-024-74065-5

[50] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar,

J. Foerster, S. Whiteson, QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 4295–4304.
URLhttps://proceedings.mlr.press/v80/ rashid18a.html