# LLM-Based Multi-Agent Decision-Making: Challenges and Future Directions

Chuanneng Sun, *Graduate Student Member, IEEE*, Songjun Huang, *Graduate Student Member, IEEE*, and Dario Pompili, *Fellow, IEEE*

***Abstract*—In recent years, Large Language Models (LLMs) have shown great abilities in various tasks, including question answering, arithmetic problem solving, and poetry writing, among others. Although research on LLM-as-an-agent has shown that LLM can be applied to Decision-Making (DM) and achieve decent results, the extension of LLM-based agents to Multi-Agent DM (MADM) is not trivial, as many aspects, such as coordination and communication between agents, are not considered in the DM frameworks of a single agent. To inspire more research on LLM-based MADM, in this letter, we survey the existing LLM-based single-agent and multi-agent decision-making frameworks and provide potential research directions for future research. In particular, we focus on the cooperative tasks of multiple agents with a common goal and communication among them. We also consider human-in/on-the-loop scenarios enabled by the language component in the framework.**

***Index Terms*—Multi-agent system, natural language models, robotics.**

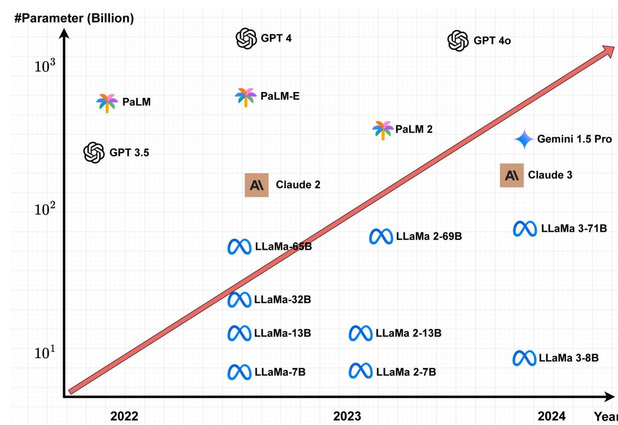


Fig. 1. Well-known large language models (LLMs) over the past three years. Among them, only PaLM-E from Google is trained specifically for embodied applications.

## I. Introduction

MULTI-AGENT Decision-Making (MADM) plays important roles in many real-world Multi-Agent Systems (MAS). As opposed to individual Decision-Making (DM)-based or traditional optimization-based solutions, MADM can bring scalability and robustness to uncertain and dynamic systems [1], [2], [3], [4], [5]. This improvement is largely attributed to the communication and coordination among agents inherent in MADM, where multiple agents learn and adapt their policies simultaneously while interacting within a shared environment and communicating with others. However, how and what to communicate among the agents in the MADM remains to be explored. Representative examples include MADM frameworks that learn to generate numerical messages using neural networks, formulate neural communication protocols, and learn targeted ad hoc communications. Despite the decent performance of the MAS frameworks achieved in various applications, they still underperform human experts. As a result, it is reasonable to think *why not leveraging human knowledge and human languages in MADM?*

Received 21 November 2024; accepted 13 April 2025. Date of publication 18 April 2025; date of current version 28 April 2025. This article was recommended for publication by Associate Editor E. Johns and Editor A. Faust upon evaluation of the reviewers' comments. This work was supported by NSF RTML under Award Number CCF-1937403. *(Corresponding author: Chuanneng Sun.)*

The authors are with the Department of Electrical and Computer Engineering, Rutgers University–New Brunswick, New Brunswick, NJ 08901-8554 USA (e-mail: chuanneng.sun@rutgers.edu; songjun.huang@rutgers.edu; pompili@rutgers.edu).

Digital Object Identifier 10.1109/LRA.2025.3562371

As recent advances in Natural Language Processing (NLP) demonstrate great abilities in multi-modal tasks, language-conditioned MADM becomes a promising research problem. NLP has been an active research topic for decades, and many famous models have been proposed for language modeling, such as Recurrent Neural Network (RNN) [6], Long-Short Term Memory networks (LSTM) [7], and transformers [8]. These foundational models have greatly improved the ability of machines to understand and generate human language, setting the stage for more complex applications.

In recent years, the integration of NLP with single-agent DM has led to the development of language-conditioned DM frameworks [9], [10], [11], especially as Large Language Models (LLMs) [12], [13], [14] emerged as the rising star in the community (see Fig. 1) and has been successfully applied in various fields [15], [16], [17]. There are works studying the combination of normal LMs with DM and MADM [10], [18]; however, the LMs in these works are mostly trained with the DM frameworks or used to simply generate word embedding. It is indeed true that LLM-based MADM inherits potential hallucination or factual inaccuracy issues; however, the adoption of multiple LLM agents in the multi-agent settings could mitigate this problem by each agent verifying the facts in other agents' outputs.

LLMs typically have billions of parameters, as opposed to normal language models that have much fewer parameters. These billions of LLM parameters contain general human knowledge about the world and can easily adapt to DM problems without the need for retraining. This integration not only leverages the semantic richness of a language but also allows for the dynamic adjustment of agent behaviors based on linguistic input. In particular, LLM can generate new information that it has not seen before on the basis of a few examples. For example, in Reflexion [19], the authors showed that the LLM agent could generate decent reflections on its decisions without any reward/feedback from the environment. Such capabilities are particularly valuable in multi-agent systems, where agents must coordinate and cooperate based on shared goals communicated through language. Note that this letter focuses on LLM-based MADM, as opposed to LLM-based Multi-Agent Reinforcement Learning (MARL). The difference is that Reinforcement Learning (RL)-based approaches generally require feedback from the external environment, while DM and MADM are more general as the feedback might come from the (LLM) agent itself or other agents.

Due to the need for communication and coordination, the problem of MADM becomes more complex than simply multiplying the DM of a single agent by the number of agents. As opposed to conventional MADM, LLMs-based MADM can leverage linguistic cues to facilitate inter-agent communication and collaboration, further boosting system performance. For example, agents can use shared language to negotiate roles, coordinate actions, or exchange information about the environment or their internal states, thereby aligning their objectives more effectively. This language-enhanced coordination becomes critical in complex scenarios where agents must handle ambiguous or evolving tasks that require continual communication and mutual understanding. The exploration of these capabilities opens up new possibilities for designing more intelligent and flexible multi-agent systems capable of operating in unpredictable, real-world environments.

Guo et al. [20] reviewed LLM-based multi-agent frameworks, but the emphasis of that paper was not on MADM. Unlike their paper, this letter focuses more on the MAS that tries to accomplish a task cooperatively. In addition to that, there are several surveys on the topic of MADM [21], [22], [23] and single agent LLM-based DM [24], [25], but none of them is dedicated to LLM-based MADM. Therefore, *we claim that we are among the first to provide a systematic overview of the LLM-based MADM problem and provide potential future research directions.*

The remainder of this letter is organized as follows. We first introduce the problem of MADM and provide a brief overview of conventional, i.e., non-LLM-based, MADM, and single-agent LLM-based DM, in Section II. Then, we will survey the existing LLM-based MADM frameworks in Section III. After that, we will discuss the challenges and future research directions for this field in Section IV. Finally, we will conclude the letter in Section VI.

## II. Preliminaries

We first introduce the problem of MADM in Section II-A. Then, we present conventional non-LLM-based MADM in Section II-B. Finally, we discuss LLM-based single-agent open-loop DM and close-loop DM in Section II-C.

### A. MADM Problem Definition

The problem of MADM is usually solved via MARL, which can be modeled with the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [26], an extension to a multi-agent manner of the Markov Decision Process (MDP). An MDP for $N$ agents consists of a set of states $\mathbf{s} \in \mathcal{S}$, which describes all the configurations for the participating agents, a set of actions $\mathcal{A}_1, \ldots, \mathcal{A}_N$ and a set of observations $\mathcal{O}_1, \ldots, \mathcal{O}_N$. Each agent $i$ has a policy $\boldsymbol{\pi}_i : \mathcal{O}_i \times \mathcal{A}_i \mapsto [0, 1]$ parameterized by $\theta_i$. We denote deterministic policies by $\boldsymbol{\mu}_i : \mathcal{O}_i \mapsto \mathcal{A}_i$. The environment will generate the next state based on the state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \mapsto \mathcal{S}$. Each agent will receive a reward from the environment as a function of state and action $r_i : \mathcal{S} \times \mathcal{A}_i \mapsto \mathbb{R}$ as well as an individual observation that is correlated with the state, $o_i : \mathcal{S} \mapsto \mathcal{O}_i$. Each agent tries to maximize its total expected return $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$, where $\gamma$ is a discount factor, and $T$ is the total time length. A key difference between Dec-POMDP and normal MDP is the partial observability, i.e., for one agent, the actions of other agents and the subsequent outcomes are not directly observable, thereby increasing the difficulty of solving the problem. Due to this partial observability, individual uncoordinated learning frameworks will not work well. Typical deep MADM frameworks adopt the actor-critic structure, where actors are trained to output the action given the observation, and the critics output a score to judge whether these actions are good in the long-term horizon.

Although our discussion often refers to Dec-POMDP (where partial observability is built in), it is worth noting that multi-agent cooperation remains challenging even in fully observable environments, largely due to issues such as non-stationarity (all agents learning concurrently) and the need for mutual consensus on strategies. Our focus on partial observability is meant to highlight language-enabled communication's potential to reduce uncertainty, but many other factors still drive MADM complexity.

### B. Traditional MADM

To solve the problem of Dec-POMDP, many frameworks have been proposed. These frameworks can be roughly categorized into two classes: learning-to-cooperate and learning-to-communicate.

*Learning to Coordinate:* The first kind of approach, such as QMIX [27], QTRAN [28], MADDPG [29], MAPPO [30], and many others [31], [32], [33], [34], [35], assumes that through centralized training with ideal communication, agents can learn to work with each other during the centralized training; therefore, communication is not needed during execution. In other words, these approaches expect the agents to learn to adapt to other agents' behavior patterns. These approaches can also be classified as policy-based and value-based approaches. Policy-based approaches typically adopt the actor-critic architecture, where actors are trained to make decisions, and critics approximate the long-term return and provide feedback to the actors. Value-based approaches learn optimized joint Q values given the team's observations and actions. A problem that often happens in this situation is the credit assignment problem, where the critic needs to determine the contribution of each agent to the performance.

*Learning to Communicate:* In communication-based approaches, agents are equipped with the capability to share information through various means, such as adjusting the content of the shared messages [36] or optimizing the structure of the communication network [37]. This explicit inter-agent communication facilitates coordinated strategies and is crucial in dynamic environments where conditions and objectives may frequently change [38], [39]. Effective communication enables agents to form coalitions to achieve common goals, adapt to peers' actions, and optimize collective outcomes, improving system performance in tasks ranging from cooperative manipulation to competitive strategic games [36]. Protocols for communication, often learned during training, leverage advanced techniques such as differentiable inter-agent learning algorithms, which refine communication patterns based on environmental feedback [40], [41], [42]. In addition, frameworks for learning emergent communication protocols/languages have also been proposed [43], [44]. These frameworks encourage the agents to learn a certain "language" that is understandable by other agents and encodes certain information.

### C. LLM-Based Single-Agent DM

As LLMs demonstrated their abilities in various tasks, several LLM-based DM frameworks have been proposed. These frameworks are not necessarily DM-based because many of them are open-loop. Although these open-loop approaches mimic aspects of reinforcement learning, they do not utilize feedback to adjust policies, making them distinct from traditional DM methods. We clarify that these frameworks are classified as DM systems *informed* by DM principles, rather than true DM frameworks.

*LLM's Role in DM:* LLMs can serve as policy generators, where the model directly maps observations to actions. For instance, in frameworks like Reflexion and Retroformer, LLMs dynamically generate actions or refine existing policies based on few-shot verbal feedback. This capability is particularly useful for handling ambiguous environments or scenarios where adaptability is required. In addition to generating policies, LLMs have been used as critics to evaluate the quality of actions or states. For example, frameworks like Refiner employ fine-tuned LLMs to provide feedback on policy decisions, leveraging the model's ability to interpret complex scenarios and offer high-level guidance. Another important application is *reward shaping or feedback generation*, where LLMs are used to enhance sparse or ambiguous reward signals. This approach is exemplified by works like Retroformer, which uses smaller language models to provide verbal feedback informed by task rewards, improving sample efficiency, and exploration strategies.

*Open-loop LLM-based Decision-Making:* Yao et al. [45] proposed ReAct, in which the LLM is prompted to generate "thoughts" to solve the problem given the observation, allowing the model to dynamically adjust and refine its strategies in response to changing environmental cues and task demands. Prasad et al. [46] proposed ADaPT, where LLMs learn to decompose the task into subtasks through short examples. Although these approaches can achieve decent performances in reasoning or word-based games, they are constrained by the knowledge the LLMs have and could be biased for certain problems. More importantly, the reward, one of the most important signals from the environment, is not considered.

*Closed-loop LLM-based DM:* There are also LLM-based DM frameworks that incorporate feedback for closed-loop control. Shinn et al. [19] proposed Reflexion, which uses few-shot verbal feedback to enhance DM capabilities. Reflexion processes feedback from interactions within task environments into textual summaries, which are then used to augment the model's episodic memory. Sun et al. [47] proposed RAHL, adopting hierarchical LLM-based policy and low- and high-level modular reflections for multi-episode improvement. Paul et al. [48] proposed Refiner, in which a fine-tuned LLM is used to provide feedback on policy decisions. Zhang et al. [49] introduced a framework that uses feedback from LLMs to enhance credit assignment in DM tasks. Their work targeted sparse reward environments and leveraged the rich domain knowledge available in LLMs to dynamically generate and refine reward functions. To improve sample efficiency, the authors proposed sequential, tree-based, and moving target feedback, facilitating more targeted exploration and reducing redundancy in state exploration. Yao et al. [50] proposed Retroformer, where a frozen LLM is used as the policy, while another smaller LM is trained to provide verbal feedback on the decisions based on the reward. Murthy et al. [51] proposed REX, adopting the Monte-Carlo Tree Search (MCTS) algorithm as the basis to solve problems. The Upper Confidence Bound (UCB) technique is adopted to guide the agent's exploration. Besides the aforementioned work, which uses LLMs as DM policies, multi-modal LLMs that are trained on DM tasks such as robot control (e.g., PaLM-E [52]) and models for grounding languages to actions [53], [54] have also been proposed. These models can achieve decent zero-shot performances in several robotic tasks because of their parameter scale.

*Comparison between LLM-based and Traditional DM:* Traditional DM approaches in MAS rely heavily on neural networks to model policies and critics. In contrast, LLM-based ones leverage the pre-trained capabilities of LLMs for tasks like policy generation, inter-agent communication, and reward shaping, saving time and effort for training and tuning the neural networks. Another advantage of LLM-based frameworks is the ability to include humans *in* or *on the loop* because agents' behaviors and actions are natural languages and can be interpreted by humans, and LLMs can easily incorporate human instructions as part of the prompts.

However, LLM-based MAS also introduces downsides. The computational demands of LLMs, which usually have billions of parameters, often make them unsuitable for real-time applications or resource-constrained environments, when serving as policies, but the computational problem is mitigated when LLMs are used as other roles, such as guiding the training of the policies or as evaluators. Furthermore, LLMs still suffer from *hallucination* problems, where they may generate descriptions or decisions based on fabricated or inaccurate information. This issue raises safety concerns, especially in scenarios involving collaboration between robots and humans, where an erroneous output could lead to critical failures or risks.

## III. Existing LLM-Based MADM

Although LLM-based MADM frameworks have not been widely studied, there is still some work focused on this topic.

*MADM for Problem Solving:* Huang et al. [65] introduced $\gamma$-Bench, which encompasses a variety of multi-agent games to assess these models. Their work included a detailed analysis of different versions of the GPT models, which demonstrated a systematic improvement in their game ability. This framework demonstrated the enhanced performance of newer LLM versions, such as GPT-4, and the potential to augment these models with reasoning techniques such as CoT. Liu et al. [55] proposed Dynamic LLM-Agent Network (DyLAN), a framework that studied the capabilities of LLM-agent collaborations for complex reasoning and code generation tasks. DyLAN adopts multiple agents with predefined roles (e.g., mathematician and programmer) and lets the agents discuss to obtain a final result. However, setting up the roles and interactions between LLM agents requires strong human priors and might scale to other tasks. Chen et al. [56] present a study on the dynamics of consensus-seeking in multi-agent systems driven by LLMs. The authors focused on the inter-agent negotiation processes, where each agent starts with a unique numerical state and negotiates to reach a unified consensus. They also provided insights on how different factors, such as agent personality (stubborn vs. suggestible), agent number, and network topology, influence the negotiation and consensus process. However, there are only two personalities, which might not be enough to solve complex problems. Hong et al. [63] proposed MetaGPT, where agents share messages with all other agents in a message pool, and agents can subscribe to messages related to their tasks. However, this work is designed specifically for code generation tasks, and the generalization ability to other tasks is yet to be explored. Li et al. [57] explored Theory of Mind (ToM) modeling with LLMs generating communication messages and beliefs about the environment and other agents. Albeit the application to ToM is novel, the design for integrating ToM in multi-agent DM is still naive. More exploration in this direction is needed. Li et al. [64] propose a pipeline for language-grounded MARL in which agents learn human-interpretable communication protocols. They first gather synthetic language data from LLM-based "expert" agents, then train MARL policies to align emergent messages with these grounded language representations

*MADM for Embodied Applications:* In addition to MADM frameworks for problem solving, there are also LLM-based MADM frameworks for embodied applications. Zhang et al. [58] proposed a Cooperative Embodied Language Agent (CoELA), a modular framework that integrates LLM to improve communication and collaborative DM among multiple agents. The structure includes a perception module for interpreting sensory data, a memory module for retaining and recalling environmental and task-related information, a communication module to facilitate inter-agent dialogue, a planning module for strategic DM, and an execution module for carrying out planned actions. By incorporating LLMs into the memory, communication, and planning modules, the framework enables agents to utilize natural language to improve both understanding and execution of cooperative tasks.

Kannan et al. [59] introduced SMART-LLM, a framework that integrated LLM with multi-agent robot task planning to translate high-level instructions into executable strategies for robot teams. By structuring task planning into sequential phases of decomposition, coalition formation, and allocation, SMART-LLM generates robot actions to achieve complex objectives. Their approach leveraged the cognitive processing power of LLMs to enhance the comprehension and execution capabilities of robot systems. Mandi et al. [60] introduced RoCo, a multi-robot arm collaboration framework with each arm equipped with an LLM agent. The LLM agents are responsible for coordination among agents by communicating with other LLM agents and path planning. Yu et al. [61] introduced Co-NavGPT, an LLM-based multi-agent navigation framework. However, unlike other frameworks where multiple LLMs are employed, in Co-NavGPT, only one LLM is used to assign frontiers to agents worldwide.

Guo et al. [62] studied the collaboration of multiple LLM-based agents on various tasks with a focus on communication and coordination among multiple agents. They proposed the Criticize-Reflect method with an LLM critic and an LLM coordinator. Although these embodied LLMs show great potential in their specific applications, there are two main problems. The first one lies in the gap between the perception module of the robot and the LLM's input. Currently, most frameworks translate sensor inputs into words, which may cause a loss of information and details. Secondly, the reasoning in these frameworks could be further improved to promote the performance of the model by adopting techniques such as CoT and RAHL [47]. Table I provides more details on these works. A more general comparison of different LLMs can be found in [66], [67].

In addition to LLM-based MADM, several works explored multi-agent interaction [68], [69], [70], e.g., multi-agent conversation and gaming. However, these works fall out of the MADM scope; hence, we will not use much space to describe them. These studies illustrate that, while the exploration into language-conditioned MADM is still nascent, it holds promise for advancing MAS capabilities. Using natural languages, these systems can achieve higher levels of coordination and understanding, which is essential in complex environments.

## IV. Open Research Problems

Despite the research efforts mentioned above, language-conditioned MADM is still an unexplored field with many unexplored aspects. To inspire more research in this field, we provide several research directions in this section. Since there have been many improvements for LLM-based single-agent DM, such as Graph Neural Network(GNN)-based Retrieval-Augmented Generation (RAG) [71], [72], we do not put emphasis on how to duplicate these improvements to multi-agent settings; instead we discuss unique challenges and research directions only MADM has. Specifically, we discuss four potential research directions: i) *personality-enabled cooperation* (Section IV-A), ii) *language-enabled human-in/on-the-loop frameworks* (Section IV-B), iii) *traditional MADM and LLM co-design* (Section IV-C), and iv) *safety and security in MAS* (Section IV-D). Fig. 2 also provides a more vivid demonstration of these research ideas.

TABLE I
EXISTING LLM FOR MADM FRAMEWORKS WITH AN EMPHASIS ON MULTI-AGENT COORDINATION

| Framework | Dataset/Simulator | Training | LLM | LLM Role |
|---|---|---|---|---|
| DyLAN [55] | MATH, MMLU; HumanEval | ✗ | GPT | Decision, Communication |
| Chen et al. [56] | Generated Data | ✗ | GPT | Decision |
| Li et al. [57] | Close-source simulator | ✗ | GPT | Decision, Communication, Theory of Mind |
| CoELA [58] | TDW-MAT, C-WAH | ✓ | LLaMA | Decision, Communication, Memory |
| SMART-LLM [59] | Proposed Benchmark Dataset | ✗ | GPT, LLaMA, Claude | Decision, Planning |
| RoCo [60] | RoCoBench | ✗ | GPT, Claude | Decision, Planning |
| Co-NavGPT [61] | Habitat-Matterport 3D | ✗ | GPT | Planning |
| Guo et al. [62] | VirtualHome-Social | ✗ | GPT | Decision, Communication |
| MetaGPT [63] | HumanEval, MBPP | ✗ | GPT | Code Generation, Communication |
| LangGround [64] | MPE | ✗ | GPT | Communication |

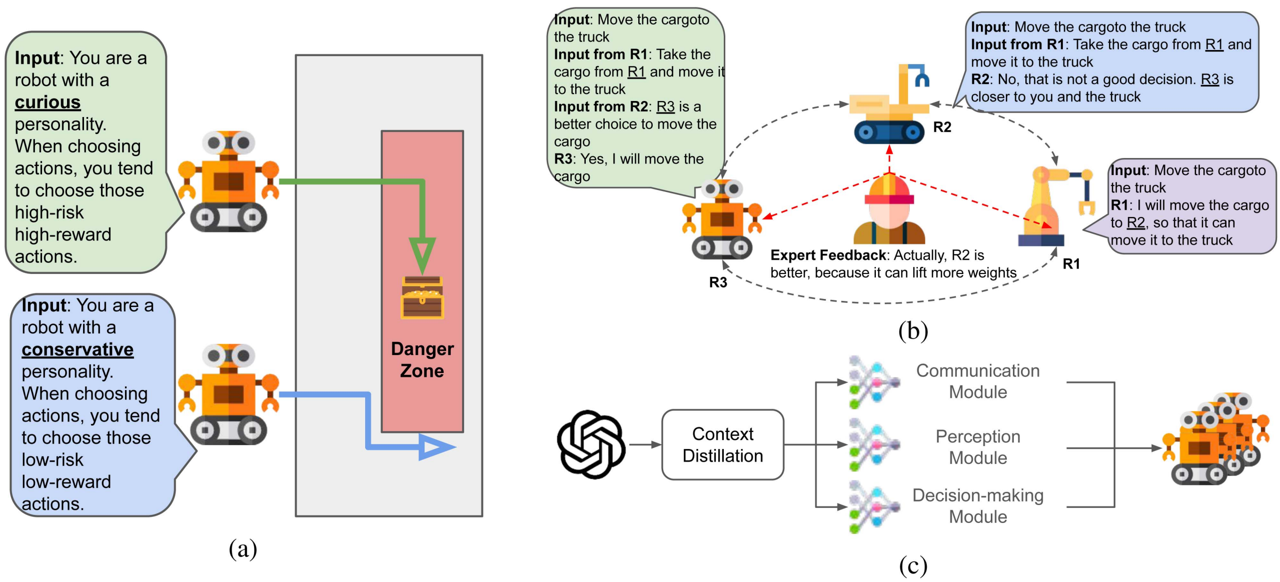


Fig. 2. Potential research directions for language-conditioned multi-agent reinforcement learning (MADM). (a) Personality-enabled cooperation, where different robots have different personalities defined by the commands. (b) Language-enabled human-on-the-loop frameworks, where humans supervise robots and provide feedback. (c) Traditional co-design of MADM and LLM, where knowledge about different aspects of LLM is distilled into smaller models that can be executed on board.

## *A. Personality-Enabled Cooperation*

Previous work [5], [56] has shown that different personalities in MADM frameworks can produce promising results. This idea can be naturally extended to language-conditioned MADM frameworks. In these frameworks, agents are distinguished by their assigned personalities. For example, an agent with a "curious" personality will tend to explore the environment, while an agent with a "conservative" personality will tend to stay in the safe areas. A team of agents with a combination of different personalities can often achieve better performance than those with the same personality. In traditional MADM frameworks, these personalities are encoded in the agents' model parameters, i.e., the weights of their models. However, with LLMs as agents, personalities can be assigned to agents by prompts, in which narratives about the agent's personality will be provided.

Another potential advantage of language-conditioned MADM with personalized agents is the ability to handle conflicts and negotiate solutions more effectively. Agents can be trained to understand and generate language-based responses that consider the perspectives and goals of other agents, facilitating a negotiation process that mirrors human interaction. This capability is particularly useful in scenarios where agents must share resources or decide on joint actions that impact the collective outcome.

However, implementing these personalized language behaviors in agents presents several challenges. The primary concern is ensuring that language models do not perpetuate or amplify undesirable biases that could lead to unfair/inefficient outcomes. Additionally, the complexity of training such models increases as they must not only understand and generate appropriate responses, but also adapt their linguistic style based on the evolving context of the interaction. Future research could focus on developing frameworks that can effectively integrate personality-driven language models into MADM systems. This integration involves creating robust prompts with memories that encode the information from past experiences in a wide range of interactive scenarios, allowing agents to learn from both their successes and failures. Furthermore, evaluating these systems will require new metrics that can assess not just the efficacy of task performance but also the appropriateness and effectiveness of communication between agents.

### B. Language-Enabled Human-in/on-the-Loop Frameworks

One of the direct advantages of language-conditioned MADM frameworks is the possibility of involving humans in or on the loop. To illustrate, human-in-the-loop frameworks [73], [74], [75] involve humans as agents that can generate actions to affect the environment, while human-on-the-loop frameworks [76] regard humans as supervisors without directly being involved in the DM process.

In human-in-the-loop setups, humans actively participate in the learning process, often providing corrective feedback or rewards to shape agent behaviors in real time. This direct interaction helps in refining the agent's actions and strategies, making them more aligned with human-like reasoning and ethical standards. For example, a human could guide an agent away from potential pitfalls in its learning process that might not be immediately apparent through algorithmic reinforcement signals alone. On the other hand, human-on-the-loop frameworks play a crucial oversight role. Here, humans monitor the system's performance and intervene only when necessary. This approach is particularly valuable in applications where autonomous operations are preferable, but human oversight is necessary to ensure safety and compliance with regulatory standards. Both of these human roles within language-conditioned MADM can benefit significantly from the integration of natural language. However, it is important to note that the humans in these tasks need to be experts in the field. The reason is that the LLMs are trained with instruction-following tasks, where they follow input instructions from humans without questioning them; therefore, when suboptimal instructions are given from non-experts, the LLM agents will tend to perform suboptimal or even unpredictable actions.

### C. Traditional MADM and LM Co-Design

Since LLMs tend to have large sizes, performing inference on-board on robot hardware is not practical. A popular way towards resource-efficient computing is through Parameter-Efficient Fine-Tuning (PEFT) techniques [77], [78], [79], [80] combined with quantization [81]. However, this kind of approach still requires inference through the large LLM network, which is impractical for small robots. To make this happen, we envision a co-design framework of traditional MADM policies and the LM models. A typical design for such systems could be to use the LLM model as a centralized critic to guide the training of the actors. This design follows the CTDE scheme introduced in Section II-B, where the critic will be removed during execution. To leverage communication during execution, we can distill the knowledge from the LLMs about communication into smaller models that can be executed onboard.

One potential development is the refinement of the distillation process, which aims to transfer knowledge from LLMs to more compact models suitable for deployment on less powerful hardware, such as robots or Internet of Things (IoT) devices. A promising direction in this direction would be in-context distillation [82], where the teacher model is an LLM with a pre-defined context.

In contrast to the co-design of LLMs and MADMs, another promising direction is the co-design of smaller LMs with MADM. Instead of scaling models to billions of parameters, employing models with significantly fewer parameters can achieve much faster inference and training speeds while maintaining acceptable accuracy. Notable examples include Bidirectional Encoder Representations from Transformers (BERT)-based models such as TinyBERT [83] and MobileBERT [84], as well as Long-Short Term Memory (LSTM)-based models like ULMFiT [85] and ELMo [86]. These models typically contain millions of parameters, as opposed to the billions found in popular LLMs. Additionally, smaller LLMs like Phi-3 [87] are demonstrating promising performance on various tasks, making them potential candidates for MAS.

### D. Safety and Security in MAS

Ensuring the safety and security of MAS is critical, especially as these systems are increasingly deployed in diverse and potentially high-stakes environments. The integration of language models into MADM introduces unique challenges and vulnerabilities, from the manipulation of agent communication to the exploitation of model biases. Many robotic operations have continuous action spaces, where the outputs of each agent's policy are continuous values. Unlike discrete action spaces, which can be reformulated as multi-choice problems and solved by prompting the multi-choice question to the LLM, continuous action space is more tricky, especially in high-stake environments, for example, operation robots. Existing methods replace the last few layers of the LLMs with new layers that map the observation in languages to continuous action spaces. However, this kind of approach requires training the new layers in the desired environment, which might be inaccessible. Therefore, exploring alternative methods for integrating LLMs into the control loop of robots operating in continuous action spaces without the need for substantial retraining or modification of the LLMs is promising.

In addition, securing the language model training process against adversarial attacks is crucial. Adversarial training, which involves exposing the system to a wide range of attack vectors during the training phase, can help models learn to resist or mitigate these attacks in deployment. In addition, input validation techniques can be employed to filter out potentially harmful or misleading inputs that could cause the system to behave unpredictably. This is particularly important in scenarios where agents interact with humans or systems outside the controlled environment and are exposed to a broader range of language inputs and behaviors.

## V. Discussion

Although LLM-based MADM holds significant promise, it is essential to continue research on traditional MADM frameworks for two key reasons. First, traditional MADM frameworks have lower computational resource requirements, making them better suited for resource-constrained robots and missions that require real-time or near-real-time decision-making, such as high-frequency trading [88], [89] and network traffic management [90], [91]. Second, current LLM-based MADM frameworks are limited to small-scale multi-agent systems due to the computational demands of LLMs and the inefficiency of communication among a large number of agents using natural language. Consequently, traditional MADM remains more suitable for large-scale applications, such as flocking [92]. Third, natural language might not be the optimal communication "language" for robot-to-robot communication because it contains irrelevant information, evidenced by studies [93] on emergent language capabilities (i.e., communicating via numerical messages) in MADM agents.

## VI. Conclusion

We provided a brief overview of Multi-Agent Decision-Making (MADM) based on conventional non-Large Language Model (LLM)-based MADM, LLM-based single-agent DM, and existing LLM-based MADM frameworks. Based on these works, we summarized potential future research directions for language-conditioned MADM, especially with the help of LLMs. Specifically, we investigated potential research directions ranging from multi-agent personality to safety and security in the LLM-based Multi-Agent System (MAS). With LLMs, designing MADM frameworks becomes more analogous to modeling the group learning process of animals or even humans, where knowledge is transferred or exchanged via natural languages. We hope, with this letter, that more research works will be enlightened and the boundary of multi-agent intelligence will be pushed further.

## References


[1] C. Sun, S. Huang, and D. Pompili, "HMAAC: Hierarchical multi-agent actor-critic for aerial search with explicit coordination modeling," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 7728–7734.

[2] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," 2016, *arXiv:1610.03295*.

[3] V. Sadhu, C. Sun, A. Karimian, R. Tron, and D. Pompili, "Aerial-DeepSearch: Distributed multi-agent deep reinforcement learning for search missions," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst.*, 2020, pp. 165–173.

[4] S. Huang, C. Sun, J. Gong, and D. Pompili, "A Bi-layer joint training reinforcement learning framework for post-disaster rescue," in *Proc. 20th Int. Conf. Distrib. Comput. Smart Syst. Internet Things*, 2024, pp. 353–360.

[5] S. Huang, C. Sun, R.-Q. Wang, and D. Pompili, "Multi-behavior multi-agent reinforcement learning for informed search via offline training," in *Proc. 20th Int. Conf. Distrib. Comput. Smart Syst. Internet Things*, 2024, pp. 19–26.

[6] M. I. Jordan, "Serial order: A parallel distributed processing approach," in *Advances in Psychology*, vol. 121. Amsterdam, The Netherlands: Elsevier, 1997, pp. 471–495.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[9] S. Peng et al., "Conceptual reinforcement learning for language-conditioned tasks," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, pp. 9426–9434.

[10] Y. Jiang, S. S. Gu, K. P. Murphy, and C. Finn, "Language as an abstraction for hierarchical deep reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32.

[11] L. Zhou and K. Small, "Inverse reinforcement learning with natural language goals," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, pp. 11116–11124.

[12] OpenAI, "ChatGPT: Optimizing language models for dialogue," 2023, Accessed: Apr. 22, 2024. [Online]. Available: https://www.openai.com/chatgpt

[13] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.

[14] A. Chowdhery et al., "PaLM: Scaling language modeling with pathways," *J. Mach. Learn. Res.*, vol. 24, no. 240, pp. 1–113, 2023.

[15] J. Wu, Z. Lai, S. Chen, R. Tao, P. Zhao, and N. Hovakimyan, "The new agronomists: Language models are experts in crop management," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 5346–5356.

[16] Z. Lai, J. Wu, S. Chen, Y. Zhou, and N. Hovakimyan, "Residual-based language models are free boosters for biomedical imaging tasks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 5086–5096.

[17] G. Han, W. Liu, X. Huang, and B. Borsari, "Chain-of-interaction: Enhancing large language models for psychiatric behavior understanding by dyadic contexts," in *Proc. IEEE 12th Int. Conf. Healthcare Inform.*, 2024, pp. 392–401.

[18] S. Havrylov and I. Titov, "Emergence of language with multi-agent games: Learning to communicate with sequences of symbols," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[19] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36, pp. 8634–8652.

[20] T. Guo et al., "Large language model based multi-agents: A survey of progress and challenges," in *Proc. 33rd Int. Joint Conf. Artif. Intell.*, 2024, pp. 8048–8057.

[21] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

[22] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Auton. Agents Multi-Agent Syst.*, vol. 33, no. 6, pp. 750–797, 2019.

[23] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," *Artif. Intell. Rev.*, vol. 55, pp. 895–943, 2022.

[24] J. Luketina et al., "A survey of reinforcement learning informed by natural language," 2019, *arXiv:1906.03926*.

[25] Y. Cao et al., "Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–21, 2024, doi: 10.1109/TNNLS.2024.3497992.

[26] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*, vol. 1. Berlin, Germany: Springer, 2016.

[27] T. Rashid, M. Samvelyan, C. S. D. Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 7234–7284, 2020.

[28] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5887–5896.

[29] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. A. I. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[30] C. Yu et al., "The surprising effectiveness of PPO in cooperative multi-agent games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 24611–24624.

[31] P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. 17th Int. Conf. Auton. Agents MultiAgent Syst.*, 2018, pp. 2085–2087.

[32] T. Rashid, G. Farquhar, B. Peng, and S. Whiteson, "Weighted QMIX: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 10199–10210.

[33] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "QPLEX: Duplex dueling multi-agent Q-learning," in *Proc. Int. Conf. Learn. Representations*, 2021.

[34] J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama, "Reducing overestimation bias in multi-agent domains using double centralized critics," 2019, *arXiv:1910.01465*.

[35] Y. Wang, B. Han, T. Wang, H. Dong, and C. Zhang, "DOP: Off-policy multi-agent decomposed policy gradients," in *Proc. Int. Conf. Learn. Representations*, 2020.

[36] J. Foerster, I. A. Assael, N. D. Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29.

[37] A. Das et al., "TarMAC: Targeted multi-agent communication," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1538–1546.

[38] S. Sukhbaatar and R. Fergus, "Learning multiagent communication with backpropagation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29.

[39] Y. Hoshen, "VAIN: Attentional multi-agent predictive modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[40] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31.

[41] I. Mordatch and P. Abbeel, "Emergence of grounded compositional language in multi-agent populations," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32.
[42] S. Shen et al., "Graphcomm: A graph neural network based method for multi-agent reinforcement learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3510–3514.
[43] S. Gupta, R. Hazra, and A. Dukkipati, "Networked multi-agent reinforcement learning with emergent communication," 2020, *arXiv:2004.02780*.
[44] A. Lazaridou and M. Baroni, "Emergent multi-agent communication in the deep learning era," 2020, *arXiv:2006.02419*.
[45] S. Yao et al., "React: Synergizing reasoning and acting in language models," in *Proc. 11th Int. Conf. Learn. Representations*, 2023.
[46] A. Prasad et al., "ADaPT: As-needed decomposition and planning with language models," in *Proc. Findings Assoc. Comput. Linguistics: NAACL 2024*, Jun. 2024, pp. 4226–4252. doi: 10.18653/v1/2024.findings-naacl.264.
[47] C. Sun, S. Huang, and D. Pompili, "Retrieval-augmented hierarchical in-context reinforcement learning and hindsight modular reflections for task planning with LLMs," 2024, *arXiv:2408.06520*.
[48] D. Paul et al., "REFINER: Reasoning feedback on intermediate representations," in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics 2024*, Malta, 2023, pp. 1100–1126.
[49] A. Zhang, A. Parashar, and D. Saha, "A simple framework for intrinsic reward-shaping for RL using LLM feedback," 2024. [Online]. Available: https://alexzhang13.github.io/assets/pdfs/Reward_Shaping_LLM.pdf
[50] W. Yao et al., "Retroformer: Retrospective large language agents with policy gradient optimization," in *Proc. 12th Int. Conf. Learn. Representations*, 2024.
[51] R. Murthy et al., "REX: Rapid exploration and exploitation for AI agents," *CoRR*, vol. abs/2307.08962, 2023, doi: 10.48550/arXiv.2307.08962.
[52] D. Driess et al., "Palm-e: An embodied multimodal language model," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 8469–8488.
[53] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 9118–9147.
[54] A. Brohan et al., "Do as I can, not as I say: Grounding language in robotic affordances," in *Proc. Conf. Robot Learn.*, 2023, pp. 287–318.
[55] Z. Liu, Y. Zhang, P. Li, Y. Liu, and D. Yang, "Dynamic LLM-agent network: An LLM-agent collaboration framework with agent team optimization," *CoRR*, vol. abs/2310.02170, 2023, doi: 10.48550/arXiv.2310.02170.
[56] H. Chen, W. Ji, L. Xu, and S. Zhao, "Multi-agent consensus seeking via large language models," 2023, *arXiv:2310.20151*.
[57] H. Li et al., "Theory of mind for multi-agent collaboration via large language models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 180–192.
[58] H. Zhang et al., "Building cooperative embodied agents modularly with large language models," in *Proc. 12th Int. Conf. Learn. Representations*, 2024.
[59] S. S. Kannan, V. L. Venkatesh, and B.-C. Min, "Smart-LLM: Smart multi-agent robot task planning using large language models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2024, pp. 12140–12147.
[60] Z. Mandi, S. Jain, and S. Song, "RoCo: Dialectic multi-robot collaboration with large language models," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 286–299.
[61] B. Yu, H. Kasaei, and M. Cao, "Co-NavGPT: Multi-robot cooperative visual semantic navigation using large language models," 2023, *arXiv:2310.07937*.
[62] X. Guo et al., "Embodied LLM agents learn to cooperate in organized teams," in *Proc. Lang. Gamification - NeurIPS 2024 Workshop*, 2024. [Online]. Available: https://openreview.net/forum?id=VKlrzygQlT
[63] S. Hong et al., "MetaGPT: Meta programming for multi-agent collaborative framework," in *Proc. 12th Int. Conf. Learn. Representations*, 2023.
[64] H. Li et al., "Language grounded multi-agent reinforcement learning with human-interpretable communication," in *Proc. Adv. Neural Inf. Process. Syst.*, 2025, vol. 37, pp. 87908–87933.
[65] J.-T. Huang et al., "How far are we on the decision-making of LLMs? Evaluating LLMs' gaming ability in multi-agent environments," *CoRR*, vol. abs/2403.11807, 2024, [Online]. doi: 10.48550/arXiv.2403.11807.
[66] T. Gao, J. Jin, Z. T. Ke, and G. Moryoussef, "A comparison of deepseek and other LLMs," 2025, *arXiv:2502.03688*.
[67] M. Steyvers et al., "What large language models know and what people think they know," *Nature Mach. Intell.*, vol. 7, pp. 221–231, 2025.
[68] Q. Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," in *Proc. ICLR 2024 Workshop Large Lang. Model Agents*, 2024.
[69] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proc. 36th Annu. ACM Symp. User Interface Softw. Technol.*, 2023, pp. 1–22.
[70] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "Camel: Communicative agents for "mind" exploration of large language model society," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36, pp. 51991–52008.
[71] D. Liu, R. Waleffe, M. Jiang, and S. Venkataraman, "GraphSnapShot: Graph machine learning acceleration with fast storage and retrieval," *CoRR*, vol. abs/2406.17918, 2024, doi: 10.48550/arXiv.2406.17918.
[72] D. Liu and M. Jiang, "Distance recomputator and topology reconstructor for graph neural networks," 2024, *arXiv:2406.17281*.
[73] D. Abel, J. Salvatier, A. Stuhlmüller, and O. Evans, "Agent-agnostic human-in-the-loop reinforcement learning," 2017, *arXiv:1701.04079*.
[74] H. Liang, L. Yang, H. Cheng, W. Tu, and M. Xu, "Human-in-the-loop reinforcement learning," in *Proc. Chin. Automat. Congr.*, 2017, pp. 4511–4518.
[75] B. Luo, Z. Wu, F. Zhou, and B.-C. Wang, "Human-in-the-loop reinforcement learning in continuous-action space," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 15735–15744, Nov. 2024.
[76] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proc. Neural Inf. Process. Syst.*, 2017, vol. 30.
[77] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Representations*, 2022.
[78] Y. Xin, J. Du, Q. Wang, K. Yan, and S. Ding, "MmAP: Multi-modal alignment prompt for cross-domain multi-task learning," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, pp. 16076–16084.
[79] Y. Xin, J. Du, Q. Wang, Z. Lin, and K. Yan, "VMT-adapter: Parameter-efficient transfer learning for multi-task dense scene understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, pp. 16085–16093.
[80] Y. Xin et al., "Parameter-efficient fine-tuning for pre-trained vision models: A survey," 2024, *arXiv:2402.02242*.
[81] D. Liu, M. Jiang, and K. Pister, "LLMEasyQuant–an easy to use toolkit for LLM quantization," 2024, *arXiv:2406.19657*.
[82] Y. Huang, Y. Chen, Z. Yu, and K. McKeown, "In-context learning distillation: Transferring few-shot learning ability of pre-trained language models," 2022, *arXiv:2212.10670*.
[83] X. Jiao et al., "TinyBERT: Distilling BERT for natural language understanding," in *Proc. Findings Assoc. Comput. Linguistics: EMNLP*, Nov. 2020, pp. 4163–4174.
[84] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: A compact task-agnostic BERT for resource-limited devices," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 2158–2170.
[85] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Vol. 1: Long Papers)*, Melbourne, VIC, Australia Jul. 2018, pp. 328–339.
[86] M. E. Peters et al., "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol., Vol. 1 (Long Papers)*, New Orleans, LA, USA, Jun. 2018, pp. 2227–2237.
[87] M. I. Abdin et al., "Phi-3 technical report: A highly capable language model locally on your phone," *CoRR*, vol. abs/2404.14219, 2024, doi: 10.48550/arXiv.2404.14219.
[88] P. Kumar, "Multi-agent deep reinforcement learning for high-frequency multi-market making," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2023, pp. 2409–2411.
[89] Y. Zheng and Z. Ding, "Reinforcement learning in high-frequency market making," *CoRR*, vol. abs/2407.21025, 2024, doi: 10.48550/arXiv.2407.21025.
[90] X. Zhao, C. Wu, and F. Le, "Improving inter-domain routing through multi-agent reinforcement learning," in *Proc. IEEE INFOCOM 2020-IEEE Conf. Comput. Commun. Workshops*, 2020, pp. 1129–1134.
[91] C. Sun, G. Jung, T. X. Tran, and D. Pompili, "Cascade reinforcement learning with state space factorization for O-RAN-based traffic steering," in *Proc. 21st Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, 2024, pp. 1–9, doi: 10.1109/SECON64284.2024.10934854.
[92] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Proc. Conf. Robot Learn.*, 2020, pp. 671–682.
[93] M. Chafii, S. Naoumi, R. Alami, E. Almazrouei, M. Bennis, and M. Debbah, "Emergent communication in multi-agent reinforcement learning for future wireless networks," *IEEE Internet Things Mag.*, vol. 6, no. 4, pp. 18–24, Dec. 2023.