Multi-Agent Recurrent Deterministic Policy Gradient with Inter-Agent Communication

1st Joohyun Cho
Dept. of ECE
Univ. of Utah
joohyun.cho@utah.edu

2nd Mingxi Liu
Dept. of ECE
Univ. of Utah
mingxi.liu@utah.edu

3rd Yi Zhou
Dept. of ECE
Univ. of Utah
yi.zhou@utah.edu

4th Rong-Rong Chen
Dept. of ECE
Univ. of Utah
rchen@ece.utah.edu

2023 57th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-2574-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/IEEECONF59524.2023.10477063

***Abstract*—In this paper, we introduce a novel approach to multi-agent coordination under partial state and observation, called Multi-Agent Recurrent Deterministic Policy Gradient with Differentiable Inter-Agent Communication (MARDPG-IAC). In such environments, it is difficult for agents to obtain information about the actions and observations of other agents, which can significantly impact their learning performance. To address this challenge, we propose a recurrent structure that accumulates partial observations to infer the hidden information and a communication mechanism that enables agents to exchange information to enhance their learning effectiveness. We employ an asynchronous update scheme to combine the MARDPG algorithm with the differentiable inter-agent communication algorithm, without requiring a replay buffer. Through a case study of building energy control in a power distribution network, we demonstrate that our proposed approach outperforms conventional Multi-Agent Deep Deterministic Policy Gradient (MADDPG) that relies on partial state only.**

***Index Terms*—Multi Agent Reinforcement Learning, Policy Gradient, Partially Observable, Actor-Critic**

I. Introduction

In recent years, the integration of Multi-Agent Reinforcement Learning (MARL) has found application in diverse domains. From autonomous traffic management, swarm robotics to collaborative multi-robot systems, energy distribution, and the analysis of economic and social problems, these technologies face uncertainties and complexities that necessitate adaptive and intelligent decision-making. At the core of many MARL algorithms lies the policy gradient framework, a approach for optimizing the policy of an agent in enhancing the adaptability and learning capabilities of agents within multi-agent systems. Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [1], multi-agent extension of DDPG, introduces a decentralized execution policy with centralized training, enabling agents to learn joint strategies in complex, interconnected environments. In the context of MADDPG, it is noteworthy that in practice, each individual agent typically contends with access to only partially observable states in environments characterized by partial observability. This asymmetry in information availability can result in a performance decline for MADDPG when compared to scenarios where agents have access to complete state information. Addressing partial observability in Partially observable Markov decision process (POMDP) has been a longstanding challenge. Previous works [2], [3] have explored recursive structures within RL frameworks to enable agents to reason about hidden states. Furthermore, extensions [4] of recursive structures to Multi-Agent systems have been introduced. An alternative strategy to alleviate the constraints imposed by partial observability in MARL is the incorporation of communication mechanisms among agents [5], [6]. This approach serves as a pivotal means to address information asymmetry and enhance collaboration, offering a promising avenue for mitigating the challenges associated with partial observability in complex, dynamic environments.

In this work, to tackle the challenge of partial observability, we propose a combining structure of Multi Agent Recurrent Deterministic Policy Gradient (MARDPG) and differentiable inter-agent communication, termed as MARDPG-IAC, to improve the performance of the existing one-stage or multi-stage MADDPG [7]. The training process for the proposed architecture unfolds in two distinct phases. During the first phase, the recursive policy function for each agent undergoes training utilizing the MARDPG algorithm. Transitioning to the communication network training stage, where other networks remain fixed, the communication network is trained on-policy. To seamlessly integrate two diverse training types, an asynchronous update scheme has been introduced. Through numerial results from the case study involving a power network, we demonstrate the superior performance of MARDPG-IAC compared to both the one-stage and multi-stage versions.

The remainder of the paper is organized as follows: Section II provides the background of MARL under POMDP. Section III presents the architecture and algorithm of the proposed MARDPG-IAC. Section IV introduces a case study of power distribution network. In Section V, we present simulation studies of MARDPG-IAC and compare with other existing algorithms. Section VI includes conclusions and future work.

II. Background

We consider multi-agent reinforcement learning (MARL) with a decentralized markov decision process (MDP) and partially observable states, denoted as $(\mathcal{M}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R})$.

The work by Rong-Rong Chen is supported in part by NSF grants CCF-1817154 and ECCS-1824558. The work of Yi Zhou is supported in part by NSF grants CCF-2106216 and DMS-2134223.

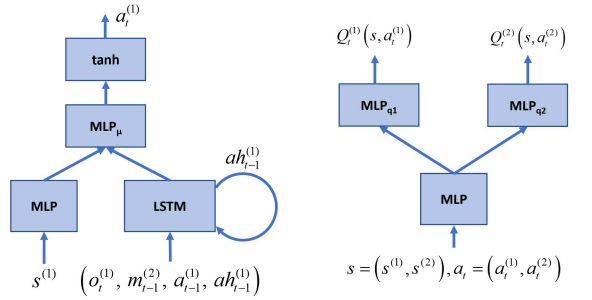
Here, $\mathcal{M}$ is a set of m agents, $\mathcal{S} = \times_i \mathcal{S}^{(i)}$ is the set of joint state space, $\mathcal{O} = \times_i \mathcal{O}^{(i)}$ is the set of joint observation space, $\mathcal{A} = \times_i \mathcal{A}^{(i)}$ is the joint action space, $\mathcal{R}$ is the reward function. Each agent i executes action $a^{(i)} \in \mathcal{A}^{(i)}$. The joint action $a = (a^{(1)}, \cdots, a^{(m)})$ causes state transition from $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ with probability $P(s'|s; a) = \mathcal{P}(s, a; s')$. Each agent i only has access to its local state $s^{(i)}$ and local observation $o^{(i)}$, and has its own policy $\mu^{(i)} : \mathcal{S}^{(i)} \times \mathcal{O}_h^{(i)} \times \mathcal{A}_h^{(i)} \to \mathcal{A}^{(i)}$ in which subscript h denotes history of agent i's observations and actions. The joint policy is denoted as $\mu = (\mu^{(1)}, \cdots, \mu^{(m)})$. The agents receive a shared joint reward of $r_{t+1} = \mathcal{R}(s_t, a_t)$ at each time $t+1$. The goal is to maximize the expected return, $J = \mathbb{E}(\sum_{t=0}^{\infty} \gamma^t r_{t+1})$, where γ is the discount factor.

Although existing RL algorithms, such as MADDPG, can be employed to tackle MDP with partial states, the efficacy of these algorithms can be substantially reduced compared to that with full states. Consequently, we propose a new approach in this work to address this performance degradation by utilizing the history of local observations, actions, and inter-agent communication. Specifically, at each time step t, we assume that each agent has access to local observations $o_t^{(i)} \in \Omega^{(i)}$, which are determined by the joint action a_{t-1} and joint state s_{t-1} from the previous time step. We incorporate a recurrent structure to accumulate this side information in our RL algorithm design, enabling agents to leverage this information to generate improved policies. It is important to note that our proposed MARL formulation differs from that of the standard decentralized partially observable MDP (Dec-POMDP). The latter assumes that each agent makes decisions based solely on local observations, whereas in our setting, local observations are employed as additional information alongside local states to facilitate the generation of better policies. Furthermore, the states in our MDP formulation cannot be construed as observations in the Dec-POMDP, as they are not necessarily a consequence of the agents' actions.

III. The Proposed MARDPG-IAC

We will begin with a description of the Modified MARDPG, followed by IAC.

Modified MARDPG: Recurrent Deterministic Policy Gradient (RDPG) [3] offers several advantages for tackling the challenges of partial observability in multi-agent environments. RDPG leverages a recurrent neural network to maintain a memory of past observations, which can be used to infer hidden state information and the actions of other agents. In this paper, we adopt modified MARDPG with recursive actor and non-recursive critic structure with centralized training and decentralized execution (CTDE) to mitigate performance degradation resulting from partial observability. In CTDE, the each critic can access full state information and all actions conducted by agents so the critic don't need to use recurrent structure to accumulate information about other agents but the actor of each agent can obtain only partail state and observation and need to take advantage of recursive structure to accumulate partial information to infer actions and partial observations by other agents. This recursive asymmetry between actor and critic structure provides accurate Temporal Difference (TD) target with less backpropagation computational burden while actors can infer information effectively. Another difference from the conventional RDPG structure is that partial state for each agent doesn't change in the recursive iteration, which means it doesn't require inference based on recursive structure. Similar to the conventional MARDPG, the modified MARDPG incorporates an actor-critic structure to facilitate policy gradient and the above described features are delineated in Figure 1. In the figure, $m_t^{(i)}$ represents the message generated by the Communication Network (CommNet) of agent i in IAC. Additionally, $ah_{t-1}^{(i)}$ and $ch_{t-1}^{(i)}$ denote the hidden states of the actor network and CommNet for agent i, respectively.



(a) actor structure for agent 1 (b) centralized critic structure

Fig. 1: Modified Recurrent Actor Critic Structure

The policy gradient with respect to agent i's policy parameterization is

$$\frac{\partial J(\theta^{(i)})}{\partial \theta^{(i)}} = \mathbb{E}_{\tau^{(i)}} \left[\sum_{t=0}^{T} \gamma^{t-1} \frac{\partial Q_\mu^{(i)}\left(s^{(1)}, \cdots, s^{(L)}, a^{(1)}, \cdots, a^{(L)}\right)}{\partial a^{(i)}} \Bigg|_{a^{(i)} = \mu_\theta^{(i)}} \frac{\partial \mu_\theta^{(i)}}{\partial \theta^{(i)}} \right]$$

where we have written the expectation over observation-action trajectory $\tau^{(i)} = (s_1^{(i)}, o_1^{(i)}, a_1^{(i)}, o_2^{(i)}, \cdots, a_{T-1}^{(i)}, o_T^{(i)})$ and $\mu_\theta^{(i)}$ is a deterministic policy with parameter $\theta^{(i)}$ for agent i and $Q_\mu^{(i)}$ is the true action-value function of agent i associated with the current policy. Figure 2 presents a schematic diagram illustrating the process of centralized critic update. The target critic generates the TD target by incorporating the current reward from the environment, and The critic network parameters are thereafter adjusted to minimize the TD error.

Inter-Agent Communication (IAC): We adopt modified Differentiable Inter-Agent Learning (DIAL) [6] structure for reinforced communication learning between agents. DIAL doesn't use experience replay to avoid non-stationarity misleading caused by multiple agents' concurrent learning and backpropagation starts once the episode reaches its terminal state or the maximum length of sequence. To apply this limitation to our model, the update for actor and the update for communication network happen asynchronously as described in Figure 3. CommNet and actor network are separated, which

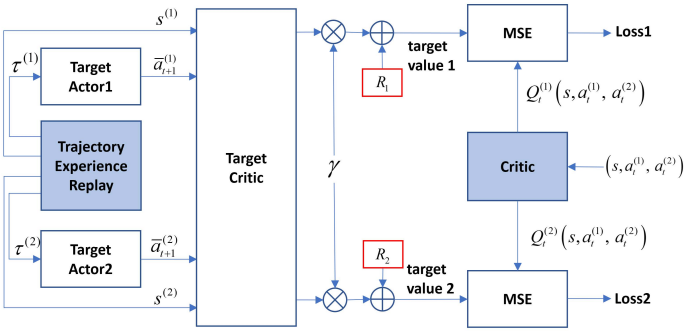
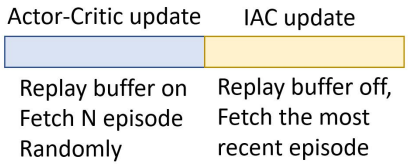


Fig. 2: MARDPG-IAC Centralized Critic Update



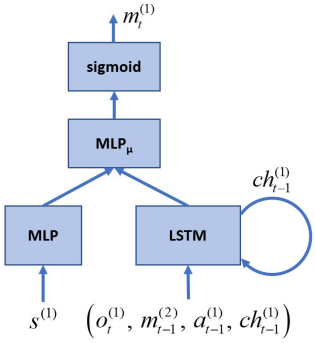


Fig. 3: Asynchronous Network Update and CommNet Structure

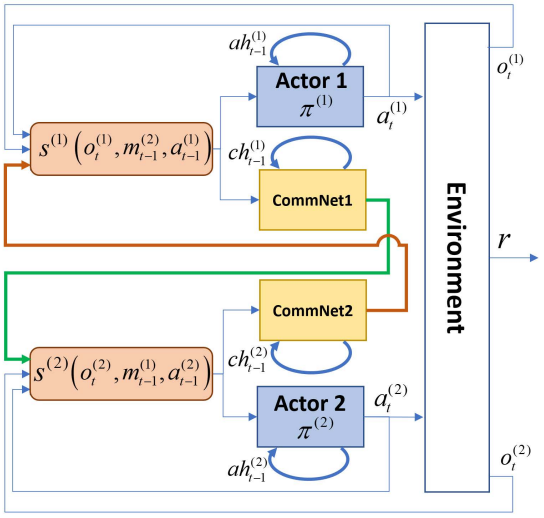


Fig. 4: MARDPG-IAC Actor and CommNet Exectution

means CommNet is not affected by the actor network's off policy update based on experience replay and while actor-critic update phase, CommNet is freezed to give stability in update. In CommNet update phase, run a episode until it reaches its terminal state or the maximum length of sequence. At the end of the trajectory, the gradient calculation for the message net begins from the time T in a backward direction. The loss function for the message network is defined by the downstream bootstrap TD error of other agents. Once CommNet has been updated, the action network update phase starts alternately. The interconnection among actor network, CommNet, and environment is described in Figure 4. A detailed description of the MARDPG-IAC is shown in Algorithm 1.

IV. A Case Study of Decentralized Building Energy Control in Power Distribution Network

To assess the efficacy of the proposed MARDPG-IAC in practical scenarios, we conduct a case study addressing a building energy control problem within a power distribution network, aimed at ensuring reliable and cost-effective grid operation. For simplicity, we assume a scenario where each node in the distribution network connects to a single building complex, allowing control over real and reactive power consumption and generation. The distribution network utilized in this study is a simplified single-phase IEEE-13 Node Test Feeder, illustrated in Figure 5.

The 13 nodes are indexed as $i = 0, \ldots, 12$, where Node 0 serves as the feeder head maintaining a constant voltage magnitude. Ensuring the reliability of grid operation necessitates maintaining voltage magnitudes within a specific range at all nodes. Let $\boldsymbol{V} \in \mathbb{R}^{12}$ represent the vector containing the voltage magnitudes of the 12 nodes (excluding the feeder head). At any given time, we have $\boldsymbol{V} = f(\boldsymbol{P}, \boldsymbol{Q})$, where the mapping $f(\cdot)$ is determined by the power distribution network's topology and configuration. Additionally, $\boldsymbol{P} = \boldsymbol{P}_b + \boldsymbol{P}_c \in \mathbb{R}^{12}$ and $\boldsymbol{Q} = \boldsymbol{Q}_b + \boldsymbol{Q}_c \in \mathbb{R}^{12}$ denote the net real and reactive power consumption vectors at the 12 buildings, with positive values indicating consumption and negative values indicating generation. In the MARDPG-IAC framework, $\boldsymbol{P}_b$ and $\boldsymbol{Q}_b$ represent the baseline net real and reactive power consumption vectors, treated as the system's *state*. Meanwhile, $\boldsymbol{P}_c$ and $\boldsymbol{Q}_c$ represent the controllable net real and reactive power consumption, considered as the system's *action*. It is noteworthy that the actions in this context are continuous-valued. The voltage magnitude V_i at the ith node is regarded as the *local observation*. Additionally, at any time, the negative total power loss of the distribution network $-L(\boldsymbol{P}, \boldsymbol{Q})$ serves as the *global reward*, while the negative generation/consumption cost $-c_i(P_{c,i}, Q_{c,i})$ of each building i is considered the *local reward*. The primary objective is to minimize the total power loss along with all local generation/consumption costs, formulated as an optimal power flow problem

$$\begin{aligned} \min_{\boldsymbol{P}_c, \boldsymbol{Q}_c} \; & L(\boldsymbol{P}, \boldsymbol{Q}) + \sum_{i=1}^{12} c_i(P_{c,i}, Q_{c,i}) \\ \text{s.t.} \; & \underline{\boldsymbol{P}}_c \le \boldsymbol{P}_c \le \overline{\boldsymbol{P}}_c; \quad \underline{\boldsymbol{Q}}_c \le \boldsymbol{Q}_c \le \overline{\boldsymbol{Q}}_c; \quad \underline{\boldsymbol{V}} \le \boldsymbol{V} \le \overline{\boldsymbol{V}}, \end{aligned} \tag{1}$$

where $\underline{\boldsymbol{P}}_c$, $\overline{\boldsymbol{P}}_c$, $\underline{\boldsymbol{Q}}_c$, and $\overline{\boldsymbol{Q}}_c$ are vectors containing local

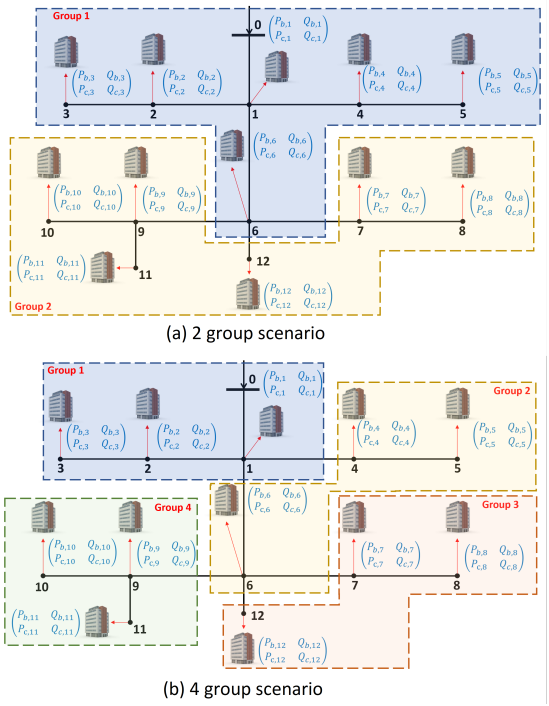


Fig. 5: Case study: Building energy control in a power distribution network.

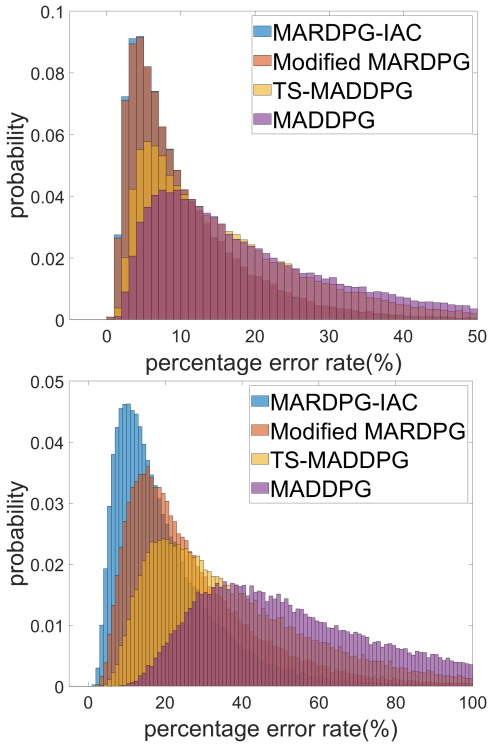


Fig. 6: Histograms of reward percentage error rate (PER) using MARDPG-IAC, Modified RDPG, TSMADDPG, and MADDPG. Left: 2 Group Scenario. Right: 4 Group Scenario

physical limits of all buildings' energy units, and $\underline{\boldsymbol{V}}$ and $\overline{\boldsymbol{V}}$ are vectors denoting the nodal voltage bounds.

V. Numerical Results

In Figure 6, The performance of the proposed MARDPG-IAC is compared with three other RL algorithms, namely Modified MARDPG without IAC, Two-Stage MADDPG (TS-MADDPG) [7], and conventional MADDPG [8], using a power distribution network described in Section 3. The power distribution network consists of 12 nodes, except the feeder head 0, which are divided into two groups of six nodes each for the scenario 1 and four groups of three nodes each for the scenario 2.

Each group of nodes is treated as an agent, and each agent has access to only the local states, partial observations and its own previous actions within their group. The performance comparison of the four RL algorithms for both scenarios are presented in Figure 6, where the left sub-figure shows the histogram of the evaluation results for scenario 1, and the right sub-figure shows the results for scenario 2. The $x-$axis represents the reward percentage error rate (PER), which is defined as the difference between the optimal reward obtained by a conventional centralized optimization algorithm and the reward obtained by applying the generated actions using each of the four RL algorithms, divided by the optimal reward. The expectation is calculated over a total of $6 \cdot 10^4$ independently generated states, where the states represent the nodal baseline power consumption/generation and are independent of each other over time. We assume that the components of each state vector follow a Gaussian distribution with zero mean and a variance of 10^6. It should be noted that the conventional algorithm assumes full knowledge of the states and the distribution network and needs to be re-run for each new state to compute the optimal reward. On the other hand, the four RL algorithms do not assume any prior knowledge of the power network topology and configuration.

Based on the results presented in Figure 6, it is evident that MARDPG-IAC and Modified MARDPG exhibit similar performance, which is notably superior to that of TS-MADDPG and MADDPG, as evidenced by their respective histograms. The primary difference between MARDPG-IAC and Modified MARDPG is the presence of IAC. This finding suggests that, for a higher percentage of states, MARDPG-IAC and Modified MARDPG can generate near-optimal actions that result in rewards that are closer to the optimal values, as indicated by a smaller reward percentage error rate (PER). In contrast, the histograms for TS-MADDPG and MADDPG exhibit heavier tails, indicating a higher probability of these algorithms failing to generate near-optimal actions compared to MARDPG-IAC and Modified MARDPG. Furthermore, the results indicate that the performance difference between MARDPG-IAC and Modified MARDPG is more pronounced in the four-agent scenario compared to the two-agent scenario. Specifically, the histogram of MARDPG-IAC in the four-agent scenario has a noticeably higher peak located closer to the left side of the graph. The reason for the performance improvement with IAC in the four-agent scenario is that, unlike the two-agent scenario, each agent in the four-agent scenario has more

hidden information to infer, and the use of a recurrent neural network alone is not sufficient to capture all the relevant information.

In summary, the results demonstrate that the incorporation of MARDPG-IAC leads to improved performance compared to MARDPG without IAC, TS-MADDPG and MADDPG, especially in scenarios with a larger number of agents, where there is more hidden information to infer.

VI. Conclusions and Future Work

This work introduces a novel MARDPG-IAC algorithm that enhances collaboration among agents and improves learning performance in MARL with partial states by utilizing history of local observations and actions as side information and inter-agent communication. The case study of power distribution network demonstrates the efficacy of MARDPG-IAC, which outperforms prior studies that only employed partial states for training optimal control policies. As a follow-up to this research, we aim to explore the impact of DDPG based Inter-Agent Communication Algorithms, such as the Attentional Communication Model (ATOC) [9], on performance without asynchronous update. Additionally, we note that the grouping topology in the power distribution network affects performance, and we expect that incorporating attention [10] to consider grouping topology can alleviate this performance dependency.

References

[1] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[2] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *2015 aaai fall symposium series*, 2015.

[3] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," *arXiv preprint arXiv:1512.04455*, 2015.

[4] J. Feng, H. Li, M. Huang, S. Liu, W. Ou, Z. Wang, and X. Zhu, "Learning to collaborate: Multi-scenario ranking via multi-agent reinforcement learning," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1939–1948.

[5] S. Sukhbaatar, R. Fergus *et al.*, "Learning multiagent communication with backpropagation," *Advances in neural information processing systems*, vol. 29, 2016.

[6] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[7] J. Cho, M. Liu, Y. Zhou, and R.-R. Chen, "Communication-free two-stage multi-agent ddpg under partial states and observations," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2021, pp. 459–463.

[8] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[9] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," *Advances in neural information processing systems*, vol. 31, 2018.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

Algorithm 1: MARDPG-IAC

Modified Multi-Agent RDPG update:

Initialize all agents' cirtic network $Q_{\omega}^{(i)}$ and actor $\mu_{\theta}^{(i)}$ with parameters ω and θ

Initialize target network $Q_{\omega'}^{(i)}$ and $\mu_{\theta}^{(i)}$ with weight $\omega' \leftarrow \omega$ and $\theta' \leftarrow \theta$

Initialize replay buffer R

for *episodes =1 to* M **do**

  initialize empty history h_0

  for *t=1 to* T **do**

    for each agent i,

    receive partial state $s_1^{(i)}$ at t=1

    receive partial observation $o_t^{(i)}$

    append partial state, previous action $a_{t-1}^{(i)}$ and partial observation to history $h_t^{(i)}$

    select action $a_t^{(i)} = \mu_{\theta}^{(i)} + \epsilon$ where ϵ is exploration noise

    receive reward r_t

  end

  Store the trajectory sequence in R

  Sample a minibatch of N trajectory episodes from R

  Compute target values for each sample episode without using recurrent network

  $$y_t^{(i)} = r_t^{(i)} + \gamma Q_{\omega'}^{(i)}\left(s_1^{(1)}, \cdots, s_1^{(L)}, \mu_{\theta}^{(1)}\left(h_t^{(1)}\right), \cdots, \mu_{\theta}^{(L)}\left(h_t^{(L)}\right)\right)$$

  Compute critic update

  $$\Delta\omega^{(i)} = \frac{1}{NT}\sum_n \sum_t \left(y_t^{(i)} - Q_w^{(i)}\left(s_1^{(1)}, \cdots, s_1^{(L)}, a_t^{(i)}\right)\right) \frac{\partial Q_w^{(i)}\left(s_1^{(1)}, \cdots, s_1^{(L)}, a_t^{(i)}\right)}{\partial \omega^{(i)}}$$

  Compute actor update

  $$\Delta\theta^{(i)} = \frac{1}{NT}\sum_n \sum_t \frac{Q_{\omega}^{(i)}\left(s_1^{(1)}, \cdots, s_1^{(L)}, \mu_{\theta}^{(1)}\left(h_t^{(1)}\right), \cdots, \mu_{\theta}^{(L)}\left(h_t^{(L)}\right)\right)}{\partial a} \frac{\mu_{\theta}^{(i)}\left(h_t^{(i)}\right)}{\partial \theta^{(i)}}$$

  Update actor and critic parameters using Adam

  Update the target networks

end

Inter Agent Communication update:

Load the last K trajectory episodes in the replay buffer R

for *episode=1 to* K **do**

  for *each episode* **do**

    Train Inter-Agent Communication Network by Modified DIAL algorithm

  end

end