Extended Summary

MULTI-AGENT RECURRENT DETERMINISTIC POLICY GRADIENT WITH INTER-AGENT COMMUNICATION (MARDPG-IAC)

Joohyun Cho, Mingxi Liu, Yi Zhou, Rong-Rong Chen
Department of Electrical and Computer Engineering,
University of Utah

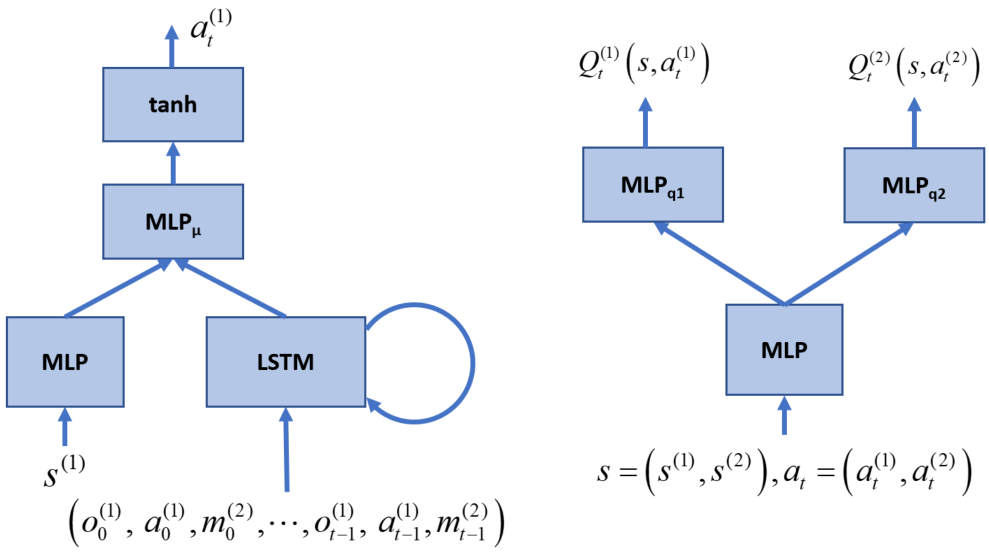
Abstract

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 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 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 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.

1 Problem Formulation

We consider multi-agent reinforcement learning (MARL) with a decentralized markov decision process (MDP) and partially observable states, denoted as $(\mathcal{M}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R})$. 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 reinforcement learning (RL) algorithms, such as multi-agent deep deterministic policy gradient (MADDPG), can be employed to tackle a Markov decision process (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 multi-agent reinforcement learning (MARL) formulation differs from that of the standard decentralized partially observable MDP (Dec-POMDP). The latter assumes that each agent makes decisions solely based on local observations, whereas in our setting, local observations are employed as additional information alongside local states to facilitate the generation of better policies.


$a_t^{(1)}$
tanh
MLP$_\mu$
MLP
LSTM
$s^{(1)}$
$\left(o_0^{(1)}, a_0^{(1)}, m_0^{(2)}, \cdots, o_{t-1}^{(1)}, a_{t-1}^{(1)}, m_{t-1}^{(2)}\right)$
$Q_t^{(1)}\left(s, a_t^{(1)}\right)$
$Q_t^{(2)}\left(s, a_t^{(2)}\right)$
MLP$_{q1}$
MLP$_{q2}$
MLP
$s = \left(s^{(1)}, s^{(2)}\right), a_t = \left(a_t^{(1)}, a_t^{(2)}\right)$

(a) actor structure

(b) critic structure

Figure 1: Agent 1's Actor-Critic Structure of Modified MARDPG for the 2 agents scenario

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.

2 The Proposed MARDPG-IAC

Modified Multi-Agent Recurrent Deterministic Policy Gradient: Recurrent Deterministic Policy Gradient (RDPG) [1] 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 structure to mitigate performance degradation resulting from partial observability. In the given system model, each critic can access full state information and all actions conducted by agents so the critic does not need to use recurrent structure to accumulate information about other agents. In contrast, the actor of each agent can obtain only partial state and observation and needs to take advantage of recursive structure to accumulate partial information to infer actions and partial observations by other agents. This asymmetry between the actor and critic structure provides accurate temperal-difference (TD) target with less backpropagation computational burden while actors can infer information effectively. Another difference from conventional RDPG structure is direct connection from fixed partial state information. During the iteration, partial state information doesn't change and contains definite information about the current partial state, which doesn't require inference based on recursive structure. The actor and critic structure adopted in the proposed structure is described in Figure 1. The message history $m_0^{(2)}, \cdots, m_{t-1}^{(2)}$ from agent 2 is fed into the recurrent structure of agent 1's actor network to generate the action $a_t^{(1)}$ at time index t.

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

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

where the expectation is over the observation-action-message trajectory $\tau^{(i)} = (s_1^{(i)}, o_1^{(i)}, a_1^{(i)}, m_1^{(i')}, o_2^{(i)}, \cdots, a_{T-1}^{(i)}, m_{T-1}^{(i')}, o_T^{(i)})$, $\mu_{\theta}^{(i)}$ 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. We replace $Q_{\mu}^{(i)}$ with the learned approximation Q^{ω} parameterized with ω for the implementation.

Inter-Agent Communication: We adopt modified DIAL (Differentiable Inter-Agent Learning) [2] 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. It is challenging to integrate the update algorithm of DIAL with that of Modified MARDPG when using an off-policy replay buffer. To combine DIAL with modified MARDPG, we use asynchronous network update scheme. The update for the action network based on modified MARDPG and the update for Inter Agent Communication(IAC) network based on DIAL happen asynchronously as described in Figure 2. The IAC network and action network are separated, which means IAC network is not affected by the action network's off policy update based on experience replay and while actor-critic update phase, IAC network is frozen to give stability in update. In IAC 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 IAC network begins from the time T in a backward direction. The loss function for agent i's IAC network is defined by the downstream bootstrap TD error of other agents $\sum_{m,t'>t} \left(\Delta M_{t+1}^{(i')} \right)$, where M is Q-network for IAC and i' is all agents indices except agent i, and the update for the agent i's IAC gradient chain is done by taking derivation of the loss function with respect to the outgoing message $m_t^{(i)}$. Once IAC network has been updated, the action network update phase starts alternately. **A detailed description of the MARDPG-IAC is shown in Algorithm 1 at the end of this document**.

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

To evaluate the proposed MARDPG-IAC in practical applications, we conduct a case study by considering a building energy control problem in a power distribution network for reliable and low-cost grid operation. For simplicity, assume each node of the distribution network is connected to only one building complex whose real and reactive power consumption and generation can be controlled. The radial power distribution network (the environment) used here is a simplified single-phase IEEE-13 Node Test Feeder, as shown in Figure 3.

Let the 13 nodes indexed by $i = 0, \ldots, 12$, where Node 0 is the feeder head maintaining a constant voltage magnitude. To ensure reliable grid operation, voltage magnitudes at all nodes should be maintained within a certain range. Let $\boldsymbol{V} \in \mathbb{R}^{12}$ denote the vector containing the voltage magnitude of all the remaining 12 nodes except the feeder head, at any time instant, we have $\boldsymbol{V} = f(\boldsymbol{P}, \boldsymbol{Q})$, where the mapping $f(\cdot)$ is determined by the power distribution network topology and configuration, and $\boldsymbol{P} = \boldsymbol{P}_b + \boldsymbol{P}_c \in \mathbb{R}^{12}$ and $\boldsymbol{Q} = \boldsymbol{Q}_b + \boldsymbol{Q}_c \in \mathbb{R}^{12}$ are the net real and reactive power consumption vectors at all 12 buildings with positive indicating consumption and negative indicating generation. In the proposed MARDPG-IAC framework, $\boldsymbol{P}_b$ and $\boldsymbol{Q}_b$ are the baseline net real and reactive power consumption vectors, which are regarded as *state*, $\boldsymbol{P}_c$ and $\boldsymbol{Q}_c$ are the controllable net real and reactive power consumption, which are considered as *action*. We note that the actions here are continuous valued. The voltage magnitude V_i, at the ith node, is considered as *local observation*. Moreover, at any time, the negative total power loss of the distribution network

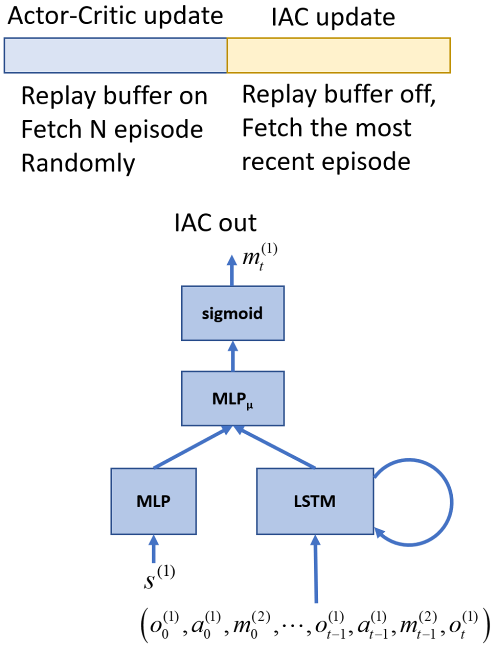
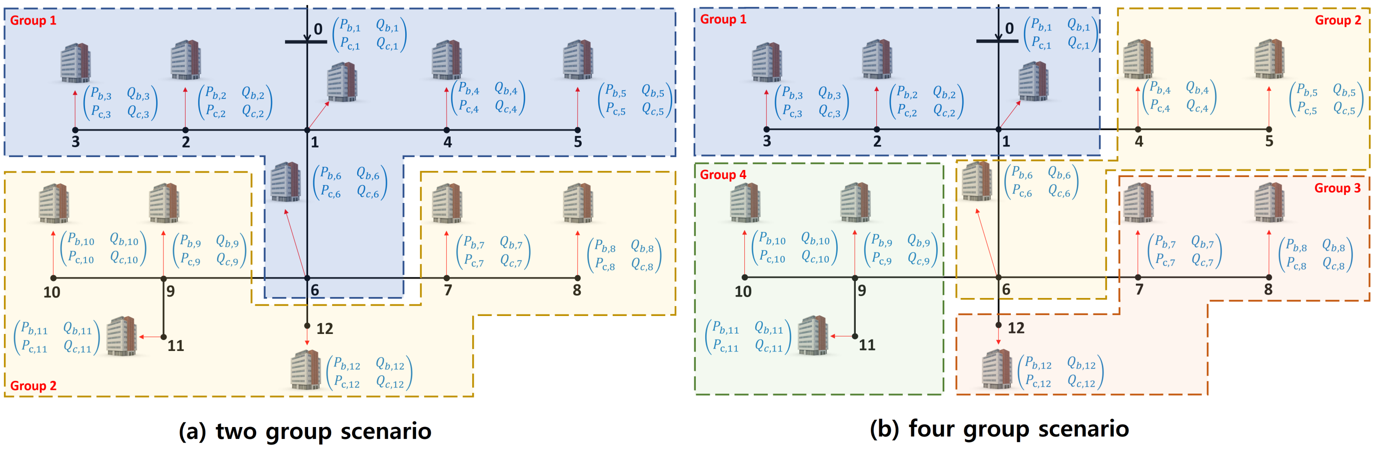


Figure 2: Asynchronous Network Update and IAC Structure of agent 1 for the 2 agents scenario


Group 1
Group 2
Group 3
Group 4
0
1
2
3
4
5
6
7
8
9
10
11
12
$\begin{pmatrix} P_{b,i} & Q_{b,i} \\ P_{c,i} & Q_{c,i} \end{pmatrix}$

(a) two group scenario

(b) four group scenario

Figure 3: Case study: Building energy control in a power distribution network.

$-L(\boldsymbol{P}, \boldsymbol{Q})$ is regarded as the *global reward*, and the negative generation/consumption cost $-c_i(P_{c,i}, Q_{c,i})$ of each building i is regarded as the *local reward*. The goal is to minimize the total power loss plus all the local generation/consumption costs, which can be formulated as the following optimal power flow problem.

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

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

4 Numerical Results

In Figure 4, The performance of the proposed MARDPG-IAC method is compared with three other reinforcement learning (RL) algorithms, namely Modified MARDPG without IAC, Two-Stage MADDPG (TS-MADDPG) [3], and conventional MADDPG [4], using a power distribution network described in Section 3. The power distribution network consists of 12 nodes, 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 4, where the left sub-figure shows the histogram of the evaluation results for scenario 1, and the right sub-figure shows the result for scenario 2. The $x-$axis of the figure 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 4, it is evident that the proposed MARDPG-IAC and Modified MARDPG exhibit similar performance, which are 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 an Inter-Agent Communication (IAC) module. 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 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.

5 Conclusions and Future Work

This work introduces a novel MARDPG-IAC algorithm that enhances collaboration among agents and improves learning performance in Multi-Agent RL 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. This work is the first to utilize history of actions and voltage observations in addition to partial states to train actor-critic networks and exhibit improved performance. As a follow-up to this research, we aim to explore the impact of DDPG Based Inter-Agent Communication

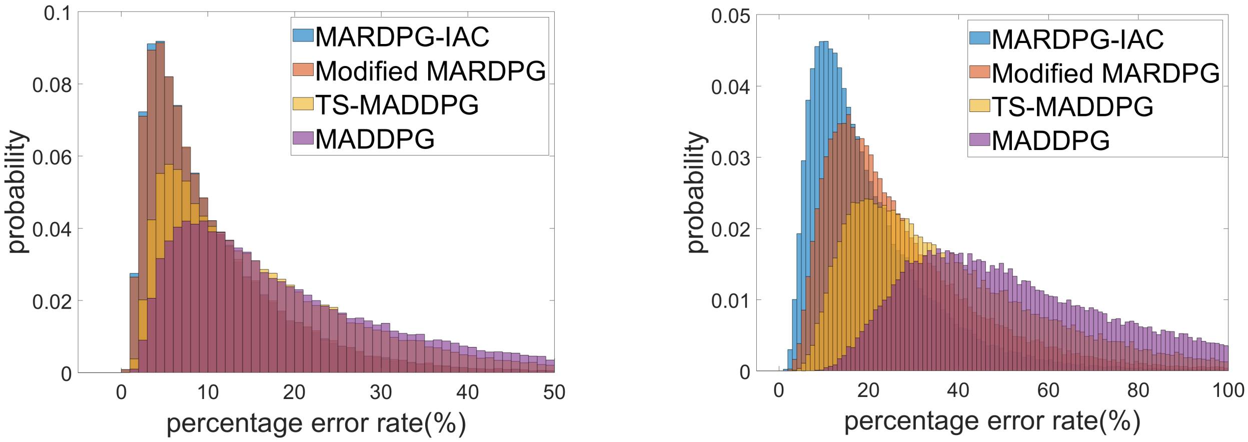


Figure 4: Histograms of reward percentage error rate (PER) using MARDPG-IAC, Modified RDPG, TSMADDPG, and MADDPG. Left: 2 Group Scenario. Right: 4 Group Scenario

Algorithms, such as the Attentional Communication Model (ATOC) [5], 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 [6] to consider grouping topology can alleviate this performance dependency.

References

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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: Multi-Agent RDPG with Inter Agent Communication under Partial States and Observations

```
Modified Multi-Agent RDPG update:
Fix IAC network parameters
Initialize all agents' cirtic network $Q_\omega^{(i)}$ and actor $\mu_\theta^{(i)}\left(h_t^{(i)}\right)$ with parameters $\omega$ and $\theta$
Initialize target network $Q_{\omega'}^{(i)}$ and $\mu_\theta^{(i)}\left(h_t^{(i)}\right)$ 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 $a_t^{(i)} = \mu_\theta^{(i)}\left(h_t^{(i)}\right) + \epsilon$ where $\epsilon$ 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:
Fix Modified MARDPG network parameters
Load the the most recent trajectory episode in the replay buffer R
for each episode do
    Train Inter-Agent Communication Network by DIAL algorithm
end
```