Hindawi
Journal of Sensors
Volume 2021, Article ID 8739206, 14 pages
https://doi.org/10.1155/2021/8739206

Research Article

Decentralized Multiagent Actor-Critic Algorithm Based on Message Diffusion

Siyuan Ding,[1] Shengxiang Li,[2] Guangyi Liu,[2] Ou Li,[2] Ke Ke,[3] Yijie Bai,[2] and Weiye Chen[2]

[1]*Key Laboratory of Experimental Physics and Computational Mathematics, China*
[2]*PLA Strategy Support Force Information Engineering University, China*
[3]*National Digital Switching System Engineering and Technological R&D Center, China*

Correspondence should be addressed to Guangyi Liu; lgy_ux@163.com

Received 2 June 2021; Accepted 16 November 2021; Published 8 December 2021

Academic Editor: Giuseppe Quero

The exponential explosion of joint actions and massive data collection are two main challenges in multiagent reinforcement learning algorithms with centralized training. To overcome these problems, in this paper, we propose a model-free and fully decentralized actor-critic multiagent reinforcement learning algorithm based on message diffusion. To this end, the agents are assumed to be placed in a time-varying communication network. Each agent makes limited observations regarding the global state and joint actions; therefore, it needs to obtain and share information with others over the network. In the proposed algorithm, agents hold local estimations of the global state and joint actions and update them with local observations and the messages received from neighbors. Under the hypothesis of the global value decomposition, the gradient of the global objective function to an individual agent is derived. The convergence of the proposed algorithm with linear function approximation is guaranteed according to the stochastic approximation theory. In the experiments, the proposed algorithm was applied to a passive location task multiagent environment and achieved superior performance compared to state-of-the-art algorithms.

1. Introduction

An agent of reinforcement learning (RL) masters skills in a trial-and-error way. The agent is settled in an environment that gives the responses and rewards corresponding to its actions over discrete time steps. The learning process is modeled as a Markov decision process (MDP) [1, 2]. The goal of the agent is to find an optimal policy that maximizes the expectation of long-term gain without the knowledge of the world (model-free [3]). Traditional tabular RL fails to handle situations with large or continuous state space, which limits its wider application. Recent theoretical and practical development has revealed that deep learning (DL) techniques with the powerful representation ability can deal with such situations efficiently. Deep reinforcement learning (DRL), a combination of RL and DL, has made remarkable achievements in the fields of chess [4], video games [5], and physical control tasks [6]. At the same time, using DRL to solve multiagent problems [7, 8] has gradually broadened so that it has now introduced a new research field, called *multiagent deep reinforcement learning* (MARL) [9]. A series of recent studies have indicated that MARL algorithms have reached the top human levels in multiplayer real-time strategy games [10]. An important problem in MARL is to learn cooperation on team tasks [7], i.e., the MARL cooperation problem. So far, the most popular solution to this problem is *centralized training and decentralized execution* [11, 12]. Methods in this fashion need a global value function [13, 14] or a global critic [11] in the training stage and assume the existence of a control center that is able to access the global state [15]. However, due to the constraints of real-world factors such as energy, geographical limitations, and communication ability, it is hard to collect the data of all agents to a center with a large number of agents.

Fully decentralized MARL algorithms yield promising results in large-scale multiagent cooperation problems, where agents learn and execute actions based on local observations. Independent Q-learning (IQL) [16, 17], as a simple

and scalability algorithm, is a typical fully decentralized MARL algorithm. An IQL agent is trained through its local observations and executes an independent local policy. However, in the perspective of an individual agent, other agents' actions are nonstationary [18], which makes the IQL fail in many tasks. In [19], the authors proposed a decentralized multiagent version of the tabular Q-learning algorithm called $\mathcal{QD}$-learning, where each agent is only aware of its own local reward, exchanges information with others, but observes the global state and joint actions. [20] assumes that all the agents are placed in a time-varying network, and it proposes a fully distributed MARL actor-critic algorithm, in which each agent has its own value function that is parameterized and updated with a weighted sum of other agents' parameters. The main limitation of this algorithm is that it also assumes that the global state and the joint actions of all agents can be obtained directly, which is still difficult to satisfy in many actual scenarios.

The problem addressed in this paper is to reduce the problem of MARL algorithms caused by centralized training. To this end, this paper proposes a model-free and completely decentralized MARL algorithm based on message diffusion. In the method, all agents are assumed to be in a time-varying communication network, where each agent obtains information from its neighbors and spreads its local observation and action in a fashion of diffusion [21]. Each agent has its own reward function, and the global reward is computed as the mean of local rewards. Each agent is designed as an actor-critic reinforcement learner and maintains a local estimation of the global state and joint actions. These estimated variables are updated with their own observations as well as messages received from neighbors, namely, in a diffusion style. We leverage stochastic approximation to analyze the convergence of the update process in the proposed algorithm with linear function approximation, and the convergence is guaranteed under reasonable assumptions. The proposed algorithm is evaluated on multiagent passive location tasks, and the results demonstrate its convergence and effectiveness.

2. Actor-Critic for a Single Agent

The reinforcement learning process can be described by an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r \rangle$, where $\mathcal{S}$ represents the set of states and $\mathcal{A}$ is the set of actions. $P(s' \mid s, a)$: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow [0, 1]$ denotes the state transition probability. The reward function is denoted as $R(s, a)$: $\mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$, and the instant reward at t is $r_{t+1} = R(s_t, a_t)$. The agent's policy is represented as $\pi : \mathcal{S} \times \mathcal{A} \longrightarrow [0, 1]$, which is parameterized to $\pi(\cdot, \cdot\,; \theta)$ or π_θ for short, with θ representing the parameters. The task of the agent is to learn a policy π_θ to maximize the expectation of the cumulative reward, i.e., the objective function

$$J(\theta) = \lim_T \mathbb{E}\left(\sum_{t=0}^{T-1} \frac{r_{t+1}}{T}\right) = \sum_{s\in\mathcal{S}} d_\theta(s) \sum_{a\in\mathcal{A}} \pi_\theta(s, a) R(s, a), \quad (1)$$

where $d_\theta(s) = \lim_t P(s_t = s \mid \pi_\theta)$ represents the stationary state distribution of the Markov chain for policy π. For policy π_θ, the action value function is defined by

$$Q_\theta(s, a) = \mathbb{E}\left\{\sum_t [r_{t+1} - J(\theta)] \mid s, a\,; \pi_\theta\right\}, \quad (2)$$

which represents the expectation of the cumulative reward in the future since state s and taking action a. $Q_\theta(s, a)$ is parameterized as $Q_\theta(s, a\,; \omega)$ or $Q_\theta(\omega)$ for short, with ω denoting the parameters of the action value function. The state value function is defined based on $Q_\theta(s, a\,; \omega)$, i.e., $V_\theta(s) = \sum_{a\in\mathcal{A}} \pi_\theta(s, a) Q_\theta(s, a)$, and the advantage function is $A_\theta(s, a) = Q_\theta(s, a) - V_\theta(s)$, which can be regarded as a measure of advantage of taking action a over other actions. According to [22], the gradient of the objective function $J(\theta)$ can be computed by

$$\nabla J(\theta) = E_{s\sim d_\theta, a\sim\pi_\theta}\{\nabla_\theta \log \pi_\theta(s, a) A_\theta(s, a)\}. \quad (3)$$

The policy parameters are updated with the policy gradient:

$$\theta \longleftarrow \theta + \beta \nabla_\theta J(\theta), \quad (4)$$

where $\beta > 0$ is the step size, and $\psi_\theta = \nabla_\theta \log \pi_\theta(s, a)$ is referred to as the *score function* of policy π_θ.

3. Agents in Time-Varying Networks

Definition 1 (Networked agents). Consider a finite directed graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{C}_t)$, where $\mathcal{V}$ and $\mathcal{C}_t$ represent the set of agents and the set of communication relations, respectively. The number of agents is $N = |\mathcal{V}|$, and the set of communication pairs is denoted as $\mathcal{C}_t = \{(i, j) \mid i \neq j, i, j \in \mathcal{V}\}$, with $C_t = [c_t(i, j)]_{N\times N}$ denoting the connection matrix of agents. The multiagent Markov decision process with networked agents is denoted by $(\mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, \mathcal{P}, \{R^i\}_{i=1}^N, \{\mathcal{G}_t\}_{t\geq 0})$. There is at least one path for any pair of agents, and each agent can obtain the observations and actions of neighbors.

At time t, the observation of agent i is $s_t^i \in \mathcal{S}^i$, where $\mathcal{S}^i \subseteq \mathbb{R}^{K_s}$ represents the observation space and $K_s \in \mathbb{N}_+$ refers to the dimension of the observation vector. The observations of all agents contribute to the global state $s_t = (s_t^1, \cdots, s_t^N) \in \mathcal{S} \subseteq \mathbb{R}^{K_s N}$, $\mathcal{S} = \prod_{i=1}^N \mathcal{S}^i$. The action of agent i is defined as $a_t^i \in \mathcal{A}^i \subseteq \mathbb{R}^{K_a}$, where $K_a \in \mathbb{N}_+$ represents the dimension of the action space. The joint actions of all agents can be expressed as $a_t = (a_t^1, \cdots, a_t^N) \in \mathcal{A} \subseteq \mathbb{R}^{K_a N}$, $\mathcal{A} = \prod_{i=1}^N \mathcal{A}^i$. Agent i holds a local estimation of the global state $s_{t,i} \in \mathcal{S}$ and a local estimation of joint actions $a_{t,i} \in \mathcal{A}$ (if t is not emphasized, they are denoted as s_i and a_i, respectively). The reward obtained by agent i is denoted by r_{t+1}^i, which is assumed to be bounded. The joint policy, $\pi : \mathcal{S} \times \mathcal{A} \longrightarrow [0, 1]$, maps the global state onto the joint actions of all agents. Actions among agents are independent, and the relationship of the joint policy and local policies satisfies

$$\pi_\theta(s_t, a_t) = \prod_{i\in\mathcal{V}} \pi_{\theta^i}(s_{t,i}, a_{t,i}), \quad (5)$$

where $\pi_{\theta^i}(\cdot,\cdot)$: $\mathcal{S}\times\mathcal{A}\longrightarrow[0,1]$ is the local policy of agent i, which is parameterized by $\theta^i\in\Theta^i\subseteq\mathbb{R}^K$, with $K\in\mathbb{N}_+$ being the dimension of agent i's parameter space. The policy parameters of all agents are represented as $\theta=[(\theta^1)^T,\cdots,(\theta^N)^T]^T\in\Theta$, where $\Theta=\prod_{i=1}^N\Theta^i$. The set of agent i's neighbors is defined by $\mathcal{N}(i)\triangleq\{j\mid j\in\mathcal{V},(i,j)\in\mathcal{E}\}$.

Assumption 2. For any $i\in\mathcal{V}$, $s_i\in\mathcal{S}$, and $a_i\in\mathcal{A}$, the policy function $\pi_{\theta^i}(s_i,a_i)>0$ is continuously differentiable at $\theta^i\in\Theta^i$. For any $\theta\in\Theta$, the Markov chain $\{s_t\}_{t\geq0}$ induced by $\pi_\theta(\cdot,\cdot)$ is irreducible and aperiodic.

The policy $\pi_{\theta^i}(\cdot,\cdot)$ is differentiable for parameter θ^i so as to use deep neural networks. Furthermore, the Markov chain induced by policy $\pi_\theta(\cdot,\cdot)$, being aperiodic and irreducible [23], has stationary distribution.

The task in the multiagent cooperation problem is to find a joint policy to maximize the expectation of the averaged long-term reward over all agents:

$$\begin{aligned}\underset{\theta}{\text{maximize}}\,J(\theta)&=\lim_T\frac{1}{T}\mathbb{E}\left(\sum_{t=0}^{T-1}\frac{1}{N}\sum_{i\in\mathcal{V}}r_{t+1}^i\right)\\&=\sum_{s\in\mathcal{S}}d_\theta(s)\sum_{a\in\mathcal{A}}\pi_\theta(s,a)\bar{R}(s,a),\end{aligned}\tag{6}$$

where $\bar{R}=(1/N)\sum R^i(s,a)$ represents the global averaged reward function, and its value at time t takes $\bar{r}_t=(1/N)\sum_{i\in\mathcal{V}}r_t^i$. Then, the global action value function Q is defined by

$$Q_t(s_t,a_t)=\mathbb{E}\left\{\sum_{t'\geq1}[\bar{r}_{t+t'}-J(\theta)]\mid s_t,a_t;\pi_\theta\right\}.\tag{7}$$

The local action value function Q^i of agent i is

$$Q_t^i(s_{t,i},a_{t,i})=\mathbb{E}\left\{\sum_{t'\geq1}\left[r_{t+t'}^i-J(\theta)\right]\mid s_{t,i},a_{t,i};\pi_{\theta^i}\right\},\tag{8}$$

which is parameterized as $Q_t^i(s_{t,i},a_{t,i};\omega_t^i)$. Since $V(s)=\sum_{a\in\mathcal{A}}\pi(s,a;\theta)Q(s,a;\omega)$, the local advantage function A^i is defined as

$$\begin{aligned}A^i(s_{t,i},a_{t,i})=Q_t^i\left(s_{t,i},a_{t,i};\omega_t^i\right)-\sum_{a'\in\mathcal{A}^i}\pi_{\theta^i}\left(s^i,a',a_{t,-i}\right)Q_t^i\\\cdot\left(s^i,a',a_{t,-i};\omega_t^i\right),\end{aligned}\tag{9}$$

where $a_{t,-i}$ represents the joint actions of all agents except for i.

Assumption 3. The global value function can be decomposed into a weighted sum of the local value functions over agents, namely,

$$Q\left(\left(s^1,\cdots,s^N\right),\left(a^1,\cdots,a^N\right)\right)=\sum_{i=1}^N c_iQ^i(s_i,a_i),\tag{10}$$

where c_i reflects the importance of individual action value function $Q^i(\cdot,\cdot)$. Note that the action value function $Q^i(\cdot,\cdot)$ of agent i depends only on its local observation and action.

Assumption 3 is similar to [14], under which the following theorem about the gradient of the global objective function to a single agent holds.

Theorem 4. *Under the conditions of Definition 1, Assumption 2, and Assumption 3, the gradient of the global objective function to agent i's policy is computed by*

$$\nabla_{\theta^i}J(\theta)=\mathbb{E}_{s\sim d_\theta(s),a\sim\pi_\theta(s,\cdot)}\left[\psi^iN^{-1}\sum_{j=1}^NA^j(s_{t,i},a_{t,i})\right],\tag{11}$$

where $\psi^i=\nabla_{\theta^i}\log\pi_{\theta^i}(s_{t,i},a_{t,i})$ is the score function, and A^j is the local advantage function defined by (9).

The proof of Theorem 4 follows a similar scheme to [22], and the complete proof is provided in Appendix A. Theorem 4 indicates that the gradient of global objective function $J(\theta)$ can be estimated by the agent's local score function ψ^i and local advantage function A^j. Although there is a certain deviation between the local observation of a single agent and the global state, in a time-varying network, an agent can improve the accuracy of the global state estimation by exchanging messages. In the next section, a completely decentralized multiagent reinforcement learning algorithm based on message diffusion over a communication network is proposed.

4. Distributed Actor-Critic Based on Message Diffusion

Suppose that, for a group of intelligent agents with communication abilities, each agent expects to obtain more information from other agents to estimate the value function and optimize its policy more accurately. But it can only obtain partial observations and receive messages via the communication network. In this section, a message diffusion-based distributed MARL algorithm is proposed to ensure agents gain global information and spread their observations efficiently.

In the algorithm, at time step t, each agent holds a global state estimation $s_{t,i}$ and joint action estimation $a_{t,i}$, which are updated by

$$\mu_{t+1}^i=\left(1-\beta_{\omega,t}\right)\cdot\mu_t^i+\beta_{\omega,t}\cdot r_{t+1}^i,\tag{12}$$

$$s_{t,i}\longleftarrow\mathbf{e}(i)\otimes s_t^i+\left(\left(\mathbb{1}_N-\mathbf{e}(i)\right)\otimes\mathbb{1}_{K_s}\right)\cdot s_{t,i},\tag{13}$$

$$a_{t,i}\longleftarrow\mathbf{e}(i)\otimes a_t^i+\left(\left(\mathbb{1}_N-\mathbf{e}(i)\right)\otimes\mathbb{1}_{K_a}\right)\cdot a_{t,i},\tag{14}$$

$$s_{t+1,i} = \sum_{j \in \mathcal{N}(i)} c_t(i,j) s_{t,j},\ a_{t+1,i} = \sum_{j \in \mathcal{N}(i)} c_t(i,j) a_{t,j}, \tag{15}$$

where μ_t^i is an estimation of the long-term return of the agent i with the bootstrapping update, and $\beta_{\omega,t} > 0$ is the step size. The agent i's global state estimation is updated in two steps. In step one, the corresponding part of the global state estimation is replaced according to (13), where $\mathbf{e}(i)$ is a unit vector with the dimension of N with only element i being 1. $\otimes$ represents the Kronecker product. 1_N is an N-dimension vector with all the elements being 1, and 1_{K_s} is defined similarly. The operation "$\cdot$" in (13) and (14) is the element-wise product. In step two, the global state estimation is updated with message diffusion according to (15). The parameters in the value function of agent i are updated through the following equation:

$$\omega_{t+1}^i = \omega_t^i + \beta_{\omega,t} \delta_t^i \nabla_{\omega^i} Q_t^i\left(s_{t,i}, a_{t,i}; \omega_t^i\right), \tag{16}$$

where δ_t^i denotes the local temporal difference error of agent i, i.e., $\delta_t^i \longleftarrow r_{t+1}^i - \mu_t^i + Q_{t+1}^i(s_{t+1,i}, a_{t+1,i}; \omega_t^i) - Q_t^i(s_{t,i}, a_{t,i}; \omega_t^i)$. According to Theorem 4, the policy of agent i is improved via gradient ascend:

$$\theta_{t+1}^i = \theta_t^i + \beta_{\theta,t} \psi_t^i \frac{1}{N} \sum_{j \in \mathcal{V}} A^j(s_{t,i}, a_{t,i}), \tag{17}$$

where $\beta_{\theta,t}$ is the step size, $A^i(s_{t,i}, a_{t,i})$ is the local advantage function defined by (9), and $\psi_t^i = \nabla_{\theta^i} \log \pi_{\theta^i}(s_{t,i}, a_{t,i})$ is the score function. Algorithm 1 summarizes the steps of the proposed approach. The algorithm works in an on-policy fashion; i.e., transitions are discarded once utilized in the update of policy parameters.

5. Convergence Analysis

To analyze the convergence of Algorithm 1, we make several assumptions on the policy update and the step sizes. Under these assumptions and linear function approximations, it is shown that both the value function and the policy function in Algorithm 1 converge almost surely.

Assumption 5. For agent i, the update of policy θ^i is a projection operator $\Gamma^i : \mathbb{R}^K \longrightarrow \Theta^i \subseteq \mathbb{R}^K$. Moreover, $\Theta = \prod_{i \in \mathcal{V}} \Theta^i$ includes at least one local maximum of $J(\theta)$.

This assumption is commonly used in the analysis of transient behavior with stochastic approximation [24, 25]. In fact, this assumption is only for the convenience of analysis and is not necessary for experiments.

Assumption 6. The step sizes $\beta_{\omega,t}$ and $\beta_{\theta,t}$ satisfy $\sum_t \beta_{\omega,t} = \sum_t \beta_{\theta,t} = \infty$, $\sum_t \beta_{\omega,t}^2 + \beta_{\theta,t}^2 < \infty$, $\lim_{t \longrightarrow \infty} \beta_{\theta,t} \beta_{\omega,t}^{-1} = 0$, and $\lim_{t \longrightarrow \infty} \beta_{\omega,t} \beta_{\omega,t+1}^{-1} = 1$.

Assumption 6 is essential for stochastic approximation and other stochastic bootstrapping algorithms. It ensures that the correction of each step becomes smaller and smaller; however, the reduction speed is not too fast to converge independently without respect to the initial states.

The action value function of agent i is approximated by a linear function family $Q^i(s, a; \omega^i) = (\omega^i)^T \phi^i(s, a)$, where $\phi^i(s, a) = [\phi_1^i(s, a), \cdots, \phi_K^i(s, a)]^T \in \mathbb{R}^K$ is the feature corresponding to the state-action pair (s, a), which is uniformly bounded. The feature matrix $\Phi \in \mathbb{R}^{|S||A| \times K}$ has full column rank, and the kth column of which is denoted by $[\phi_k(s, a), s \in \mathcal{S}, a \in \mathcal{A}]^T$. For convenience, $\phi(s_i, a_i)$ is abbreviated as ϕ^i.

Theorem 7. *Under Assumptions 2, 3, 5, and 6, for any policy π_θ, the value parameter sequence $\{\omega_t^i\}$ generated by (16) converges to ω_θ^i with probability 1, where ω_θ^i is related to policy π_θ.*

Proof. With linear function approximation, the update process of parameter ω_t^i is

$$\omega_{t+1}^i = \omega_t^i + \beta_{\omega,t} \delta_t^i \phi_t^i. \tag{18}$$

The gradient $\delta\phi$ in the update process can be seen as the error caused by current parameter ω_t^i. We construct the square of it as $Z(\omega_t^i) = (1/2)(\delta\phi)^T(\delta\phi)$. When $Z(\omega_t^i)$ takes the minimum value, $\delta\phi = 0$ holds. Hence, the convergence point of ω_t^i can be found through optimizing $Z(\omega_t^i)$. The gradient of $Z(\omega_t^i)$ is $\nabla_{\omega_t^i} Z(\omega_t^i) = (\phi(\phi' - \phi)^T)^T(\delta\phi)$, where $\phi' = \phi_{t+1}$. Rewrite (16) via introducing a random variable ξ_t to estimate $\delta_t \phi_t$:

$$\begin{aligned} \xi_{t+1}^i &= \xi_t^i + \beta_{\xi,t}\left(\delta_t^i \phi_t^i - \xi_t^i\right), \\ \omega_{t+1}^i &= \omega_t^i + \beta_{\omega,t}\left(\phi_{t+1}^i - \phi_t^i\right)\left(\phi_t^i\right)^T \xi_t^i, \end{aligned} \tag{19}$$

where $\beta_{\xi,t} > 0$ is the update step size, which satisfies Assumption 6. We assume that the second-order moments of ϕ_t^i and ϕ_{t+1}^i are bounded. Let $A = \mathbb{E}[\phi_t(\phi_t - \phi_{t+1})^T]$, $b = \mathbb{E}[(r_t - \mu_t)\phi_t]$, $\nu_t = \xi_t/\sqrt{\eta}$, $\eta > 0$, $\rho_t^T = (\nu_t^T, \omega_t^T)$, and $g_{t+1}^T = ((r_t - \mu_t)\phi_t^T, 0^T)$. Rewrite the update process in (19) by

$$\rho_{t+1} = \rho_t + \beta_{\omega,t}\sqrt{\eta}(G_{t+1}\rho_t + g_{t+1}), \tag{20}$$

where

$$G_{t+1} = \begin{pmatrix} -\sqrt{\eta} I & \phi_t(\phi_{t+1} - \phi_t)^T \\ (\phi_{t+1} - \phi_t)\phi_t^T & 0 \end{pmatrix}. \tag{21}$$

Input: Initialize parameters $\mu_0^i, \omega_0^i, \theta_0^i, \forall i \in \mathcal{V}$ and step sizes $\beta_{\omega,t}, \beta_{\theta,t}$. Agent i initializes a_0^i and obtains observation s_0^i, then initializes estimations $s_{0,i}, a_{0,i}$. Initialize the counter $t \longleftarrow 0$.

1: **repeat**
2: **for** $i = 1, \cdots, N$ **do**
3: Sample and execute an action a_t^i from the policy function $\pi_{\theta^i}(s_{t,i}, \cdot)$, and obtain s_{t+1}^i, r_{t+1}^i;
4: Update the reward $\mu_{t+1}^i \longleftarrow (1 - \beta_{\omega,t})\mu_t^i + \beta_{\omega,t} r_{t+1}^i$;
5: Update the local state $s_{t,i} \longleftarrow \mathbf{e}(i) \otimes s_t^i + ((1_N - \mathbf{e}(i)) \otimes 1_{K_s}) \cdot s_{t,i}$;
6: Update the local action $a_{t,i} \longleftarrow \mathbf{e}(i) \otimes a_t^i + ((1_N - \mathbf{e}(i)) \otimes 1_{K_a}) \cdot a_{t,i}$;
7: Send $s_{t,i}$ and $a_{t,i}$ to neighbors.
8: **end for**
9: **for** $i = 1, \cdots, N$ **do**
10: Update the estimation of the global state $s_{t+1,i} \longleftarrow \sum_{j \in \mathcal{N}(i)} c_t(i,j) s_{t,j}$;
11: Update the estimation of the joint action $a_{t+1,i} \longleftarrow \sum_{j \in \mathcal{N}(i)} c_t(i,j) a_{t,j}$;
12: **end for**
13: **for** $i = 1, \cdots, N$ **do**
14: Calculate the temporal difference error
$\delta_t^i \longleftarrow r_{t+1}^i - \mu_t^i + Q_{t+1}^i(s_{t+1,i}, a_{t+1,i}; \omega_t^i) - Q_t^i(s_{t,i}, a_{t,i}; \omega_t^i)$
15: Update the critic $\omega_{t+1}^i \longleftarrow \omega_t^i + \beta_{\omega,t} \delta_t^i \nabla_{\omega^i} Q_t^i(s_{t,i}, a_{t,i}; \omega_t^i)$;
16: Update the advantage function
$A^i(s_{t,i}, a_{t,i}) = Q_t^i(s_{t,i}, a_{t,i}; \omega_t^i) - \sum_{a' \in \mathcal{A}^i} \pi_{\theta^i}(s^i, a', a_{t,-i}) Q_t^i(s^i, a', a_{t,-i}; \omega_t^i)$;
17: Update the score function $\psi_t^i \longleftarrow \nabla_{\theta^i} \log \pi_{\theta^i}(s_{t,i}, a_{t,i})$;
18: Update the actor $\theta_{t+1}^i \longleftarrow \theta_t^i + \beta_{\theta,t} \beta_{\theta,t} \psi_t^i (1/N) \sum_{j \in \mathcal{V}} A^j(s_{t,i}, a_{t,i})$;
19: **end for**
20: $t \longleftarrow t + 1$;
21: **until** the algorithm converges.

Algorithm 1: Multiagent actor-critic algorithm based on message diffusion.

We analyze the convergence of (20) with the stochastic approximation theory [26]. Let

$$G = \mathbb{E}[G_t] = \begin{pmatrix} -\sqrt{\eta} I & A \\ A^T & 0 \end{pmatrix}, \quad g = \mathbb{E}[g_t] = \begin{pmatrix} b \\ 0 \end{pmatrix}. \tag{22}$$

As $\mathbb{E}[\delta\phi] = 0$ and $G\rho + g = 0$, we rewrite (20) as

$$\begin{aligned} \rho_{t+1} &= \rho_t + \beta_{\omega,t}\sqrt{\eta}(G\rho_t + g + (G_{t+1} - G)\rho_t + (g_{t+1} - g) \\ &= \rho_t + \beta'_{\omega,t}(h(\rho_t) + M_{t+1})), \end{aligned} \tag{23}$$

where $\beta'_{\omega,t} = \beta_{\omega,t}\sqrt{\eta}$, $h(\rho) = G\rho + g$, $M_{t+1} = (G_{t+1} - G)\rho_t + g_{t+1} - g$, and $\mathcal{F}_t = \sigma(\rho_1, M_1, \cdots, \rho_{t-1}, M_t)$. The stochastic approximation theory requires the following:

(1) Function h is Lipschitz continuous, and $h_\infty(\rho) = \lim_{c \longrightarrow \infty} h(c\rho)/c$ exists

(2) (M_t, F_t) is a martingale difference sequence, and there exits $C_0 > 0$ so that $\mathbb{E}[\|M_{t+1}\|^2 \mid \mathcal{F}_t] \le C_0(1 + \|\rho_t\|^2)$ for any ρ_1

(3) The step size $\beta'_{\omega,t}$ satisfies Assumption 6

(4) Ordinary differential equation $\dot{\rho} = h(\rho)$ has origin as the unique global asymptotically stable equilibrium

The update process described by (20) obviously satisfies conditions (2) and (3). To satisfy condition (1), it only needs to judge whether $\lim_{c \longrightarrow \infty}[(r - \mu)\phi]/c$ is bounded. It is rational to let ϕ be bounded, and ϕ' is updated based on message diffusion and bounded because C_t is a stochastic matrix. According to Definition 1, the instant reward r is bounded. We proved the convergence analysis of r, which indicates that μ is bounded, in Appendix B.

Since the feature matrix ϕ has full rank, A is a nonsingular matrix. Let $G = A^T A$, and then $\det(G) = \det(A^T A) \neq 0$, and the eigenvalues of G are nonzero. Denote an eigenvalue of G by $\lambda \in \mathbb{C}$, $\lambda \neq 0$, and the corresponding feature vector by $x \in \mathbb{C}^{2K}$, $\|x\|^2 = x^* x = 1$. Let $x^T = (x_1^T, x_2 T)$, where $x_1, x_2 \in \mathbb{C}^K$. According to the definition of G, $\lambda = x^* G x = -\sqrt{\eta}\|x_1\|^2 + x_2^* A^T x_1 - x_1^* A x_2$. Because A is a real matrix, $(x_2^* A^T x_1)^* = x_1^* A x_2$. The real part of feature vector λ is $\mathrm{Re}(\lambda) = -\sqrt{\eta}\|x_1\|^2 < 0$. So $\dot{\rho} = h(\rho)$ has global asymptotically stable equilibria. According to the stochastic approximation theory [26], the update described by (21) converges; thus, $\{\omega_t^i\}$ converges almost surely. □

Theorem 8. *Under Assumptions 2, 5, and 6, the policy parameter θ^i updated through (17) converges to the asymptotically stable equilibrium of the following ordinary differential equation (ODE) with the probability of 1:*

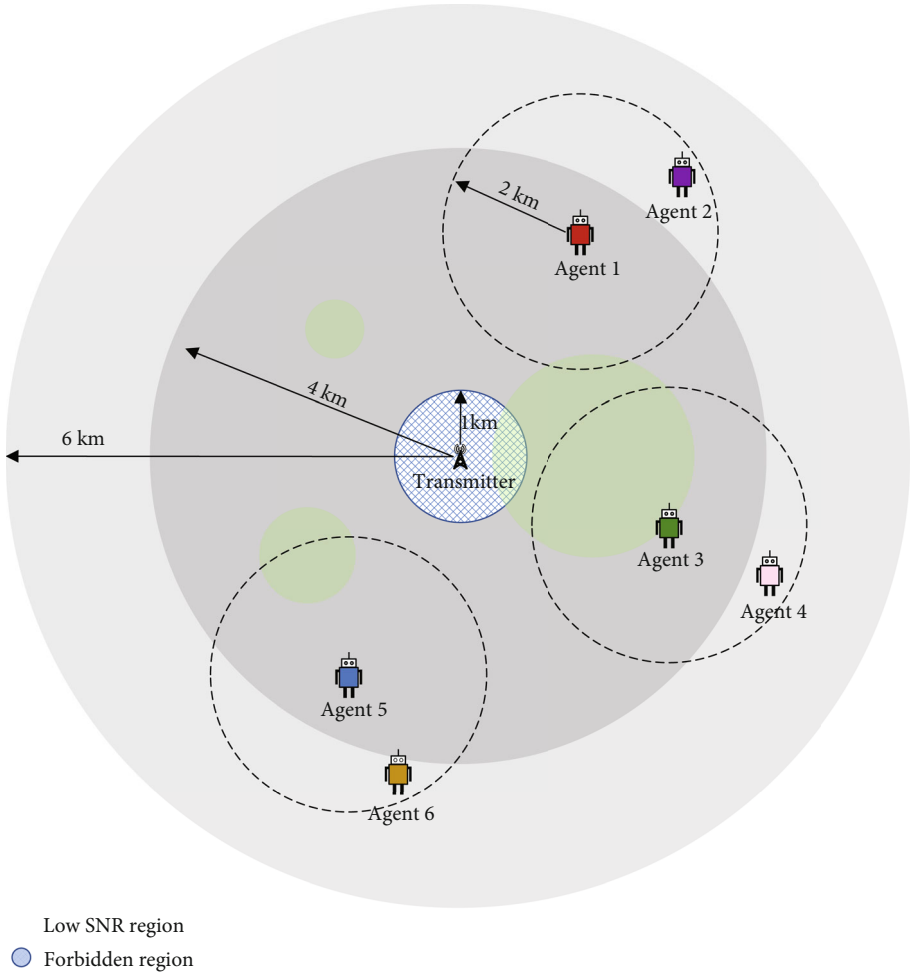


FIGURE 1: Multiagent passive location task environment. Agents need to find an optimal geometry to improve the positioning precision collaboratively. Each agent can only access partial observation, i.e., signals received by itself as well as its own position information. An agent can communicate with other agents within 2 km (neighbors), sharing individual observations. The light gray area shows the physical feasible area for all the agents, and the dark gray area shows where the agents can receive valid signals for passive positioning.

$$\dot{\theta}^i = \Gamma\wedge^i\left[\mathbb{E}_{s_t\sim d_\theta, a_t\sim\pi_\theta}\left(A_t^i\cdot\psi_t^i\right)\right],\quad \forall i\in\mathcal{V}, \tag{24}$$

where $\Gamma\wedge^i(g(\theta)) = \lim_{0<\eta\longrightarrow 0}((\Gamma^i(\theta^i+\eta g(\theta))-\theta^i)/\eta)$, *and* $g:\Theta\longrightarrow\Theta$ *is a continuous function, e.g.,* $g(\theta)=A_t^i\psi_t^i$.

Proof. Denote the σ field generated by $\{\theta_\tau, \tau\le t\}$ as $\mathcal{F}_{t,2} = \sigma(\theta_\tau, \tau\le t)$. Define two random variables:

$$\begin{aligned}\zeta_{t+1,1} &= A_t\psi_t - \mathbb{E}[A_t\psi_t\mid\mathcal{F}_{t,2}],\\ \zeta_{t+1,2} &= \mathbb{E}\left[\left(A_t - A_{t,\theta_t}\right)\psi_t\mid\mathcal{F}_{t,2}\right],\end{aligned} \tag{25}$$

where

$$A_{t,\theta_t}^i = \phi_t^T\omega_\theta - \sum_{a^i\in\mathcal{A}^i}\pi_{\theta^i}\left(s_{t,i}, a^i, a_{t,-i}\right)\phi\left(s_{t,i}, a^i, a_{t,-i}\right)^T\omega_\theta. \tag{26}$$

The update process of (17) is represented in projection as

$$\theta_{t+1}^i = \Gamma^i\left[\theta_t^i + \beta_{\theta,t}\left(\mathbb{E}\left(A_{t,\theta_t}^i\psi_t^i\mid\mathcal{F}_{t,2}\right) + \zeta_{t+1,1}^i + \zeta_{t+1,2}^i\right)\right]. \tag{27}$$

Due to the convergence of the policy evaluation, $\omega_t^i \longrightarrow \omega_\theta^i$, we have $A_t^i \longrightarrow A_{t,\theta_t}^i$. Let $M_t^i = \sum_{\tau=0}^t\beta_{\theta,t}\zeta_{t+1,1}^i$, and then $\{M_t^i\}$ is a martingale difference sequence. Since $\{\omega_t^i\}$, $\{\psi_t^i\}$, and $\{\phi_t^i\}$ are bounded, $\{M_t^i\}$ is bounded. Based on Assumption 6, it holds that

$$\sum_t\mathbb{E}\left(\|M_{t+1}^i - M_t^i\|^2\mid\mathcal{F}_{t,2}\right) = \sum\|\beta_{\theta,t}\zeta_{t+1,1}\|^2 < \infty. \tag{28}$$

Furthermore, $\{M_t^i\}$ converges according to the convergence theory of the martingale difference sequence [23]. Thus, for any $\varepsilon > 0$,

$$\lim_t\mathbb{P}\left(\sup_{n\ge t}\|\sum_{\tau=t}^n\beta_{\theta,\tau}\zeta_{\tau+1,1}^i\|>\varepsilon\right) = 0. \tag{29}$$

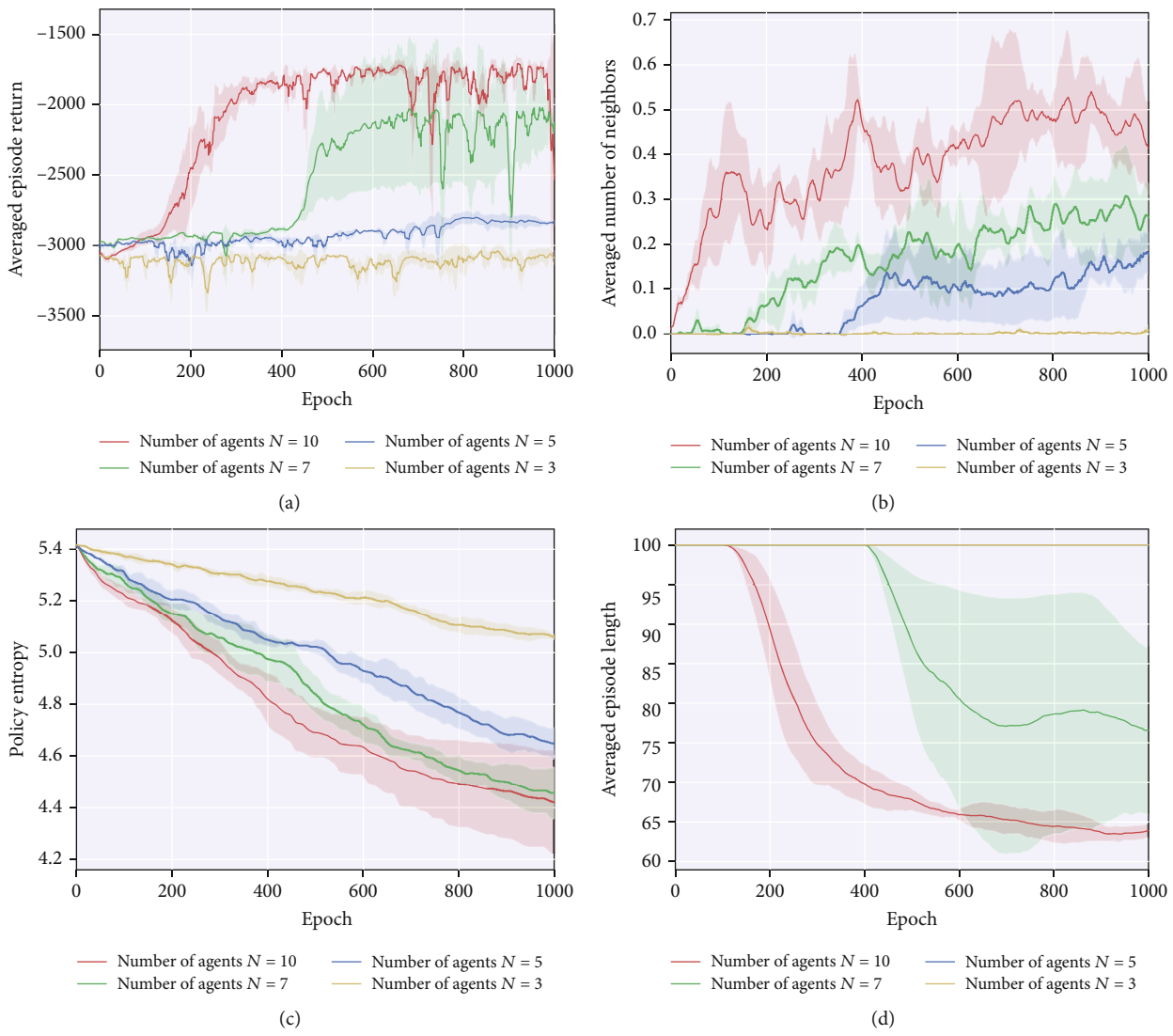


Figure 2: Training curves for the passive location tasks over different scales of agents across 5 random seeds. (a) The averaged episode return. (b) The averaged number of neighbors. (c) Mean policy entropy across all the agents. (d) Mean loss of value function across all the agents.

Therefore, (27) satisfies the necessary conditions of the Kushner-Clark theory [28, 29], and the policy parameter θ^i will converge to the asymptotically stable equilibrium of ODE (24). □

6. Experiments

In this section, from two aspects, our proposed method is evaluated:

(i) Ablation study, which investigates the influence of the agent number on the performance

(ii) Comparison study, which studies the advantages of the proposed method, compared with existing methods

6.1. Multiagent RL for Passive Location Tasks. The experiments are performed with a passive location task environment, which is a reinforcement learning environment where agents need to automatically find an optimal geometry to improve the positioning precision. The environment is introduced by [27], where all the agents are controlled by a brain that maps the global observation into joint actions. We modified the environment into a multiagent one by limiting the observation so that each agent can only access its own radio signals and position. Furthermore, each agent has a distinct brain that consists of an actor and a critic, learning and making decisions independently.

The whole scheme of the environment is shown in Figure 1. Consider a circular region with a radius of 6 km, at the center of which is a transmitter that emits radio signals all the time. The area within 1 km of the transmitter is a forbidden area, filled with the blue grid. Each agent is equipped with a radio receiver that can intercept wireless signals to estimate the position of the transmitter. All the

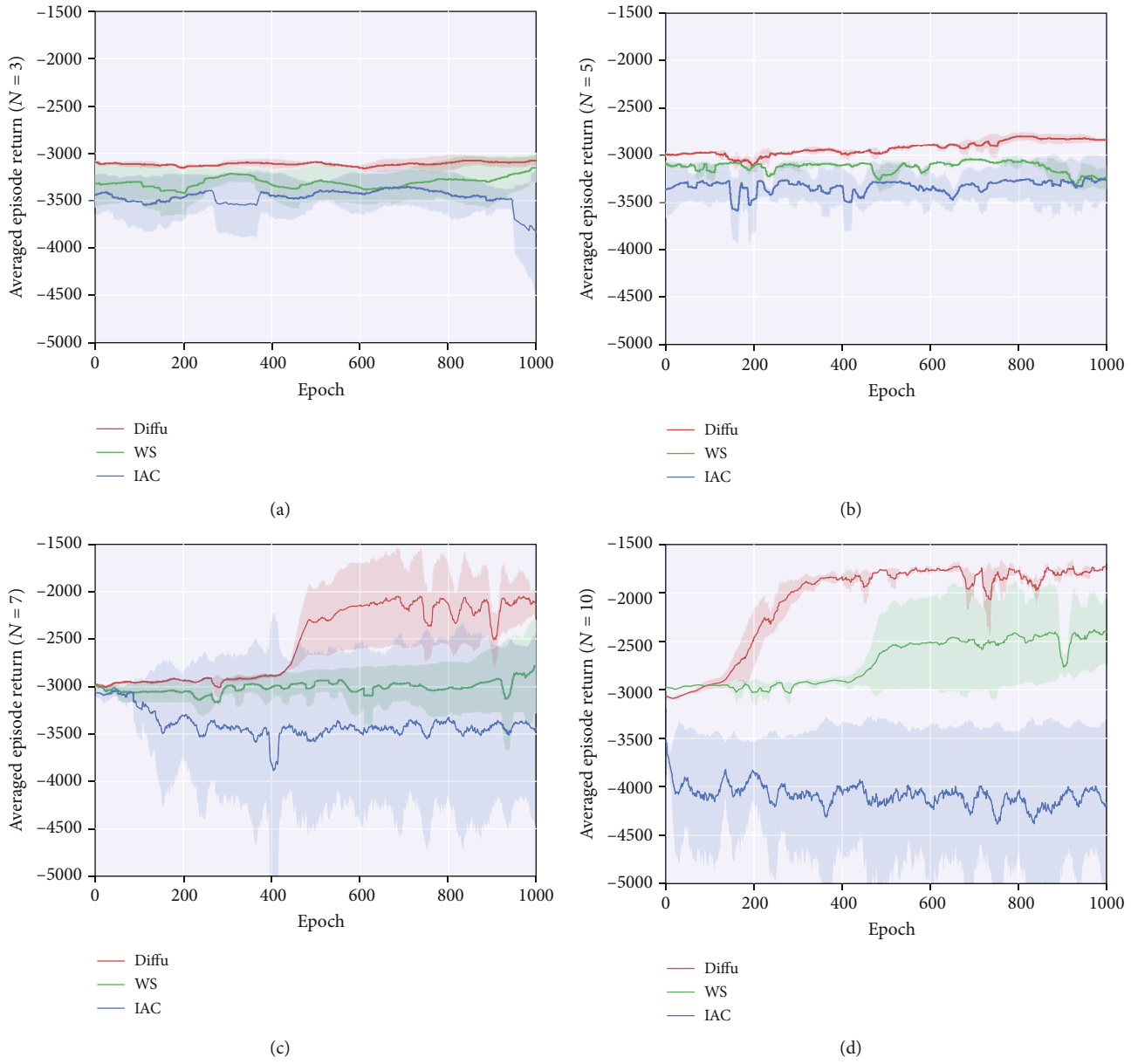
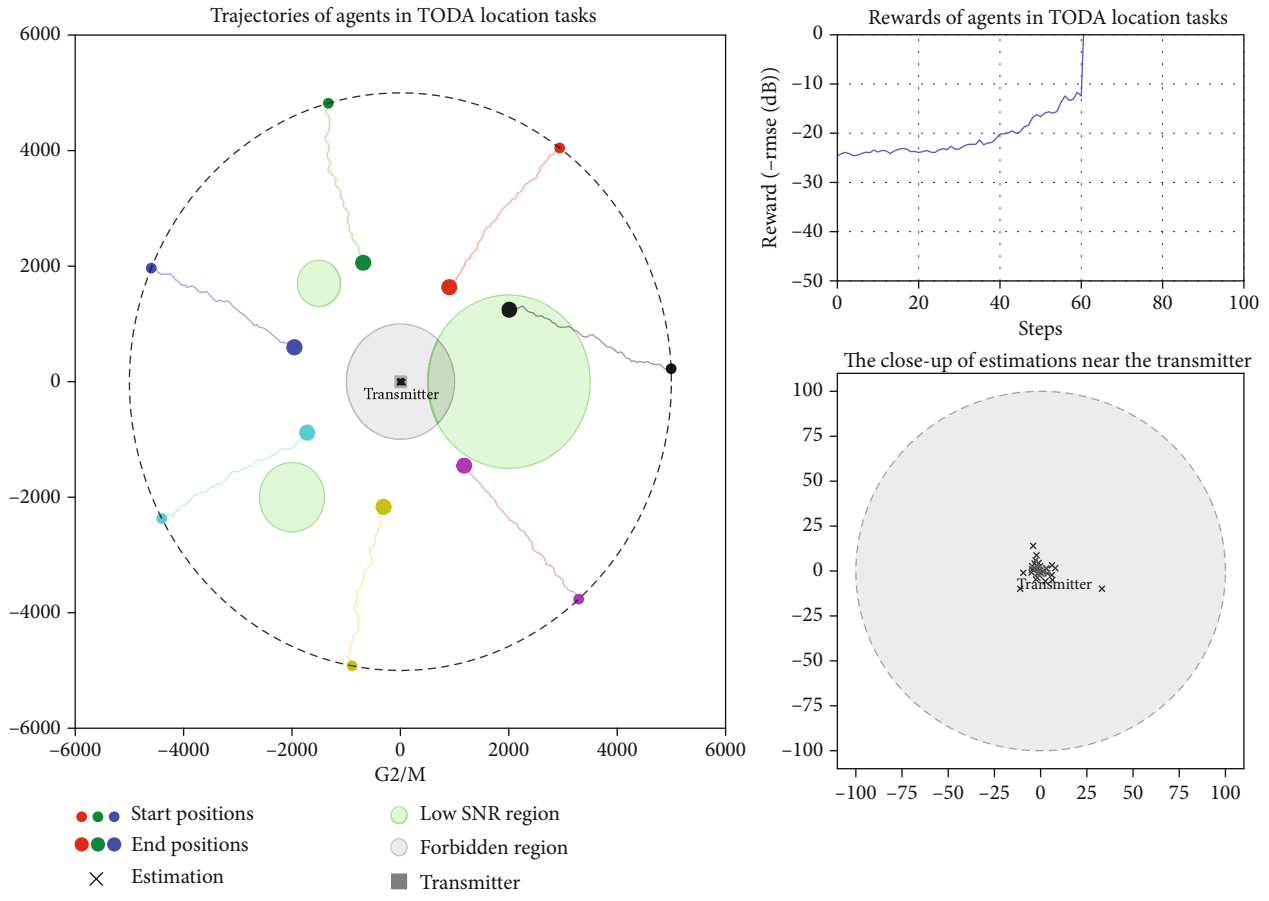


Figure 3: Performance curves compared to IAC and WS over different numbers of agents. (a) $N = 3$. (b) $N = 5$. (c) $N = 7$. (d) $N = 10$.

radio receivers are assumed to have the same performance in processing wireless signals. According to the sensitivity of the radio receiver, when agents go beyond a distance of 4 km away from the transmitter, nothing can be received. So, it is better for the agents to optimize the geometry within a closer region to the transmitter, which is shown as the dark gray area in Figure 1. Considering the multipath and interference of electromagnetic propagation, there are three low signal-to-noise ratio (SNR) regions, where signals received by the agents are contaminated by strong noises, leading to low positioning precision. The position of the transmitter is estimated in two steps: firstly, figuring out the time lag of signal propagation of each pair of agents, and secondly, estimating the transmitter's position that satisfies the time lags obtained in the first step with *least square* algorithms. The task of the agents is to navigate to an optimal geometry configuration step-by-step, avoiding low SNR and forbidden regions, improving positioning precision.

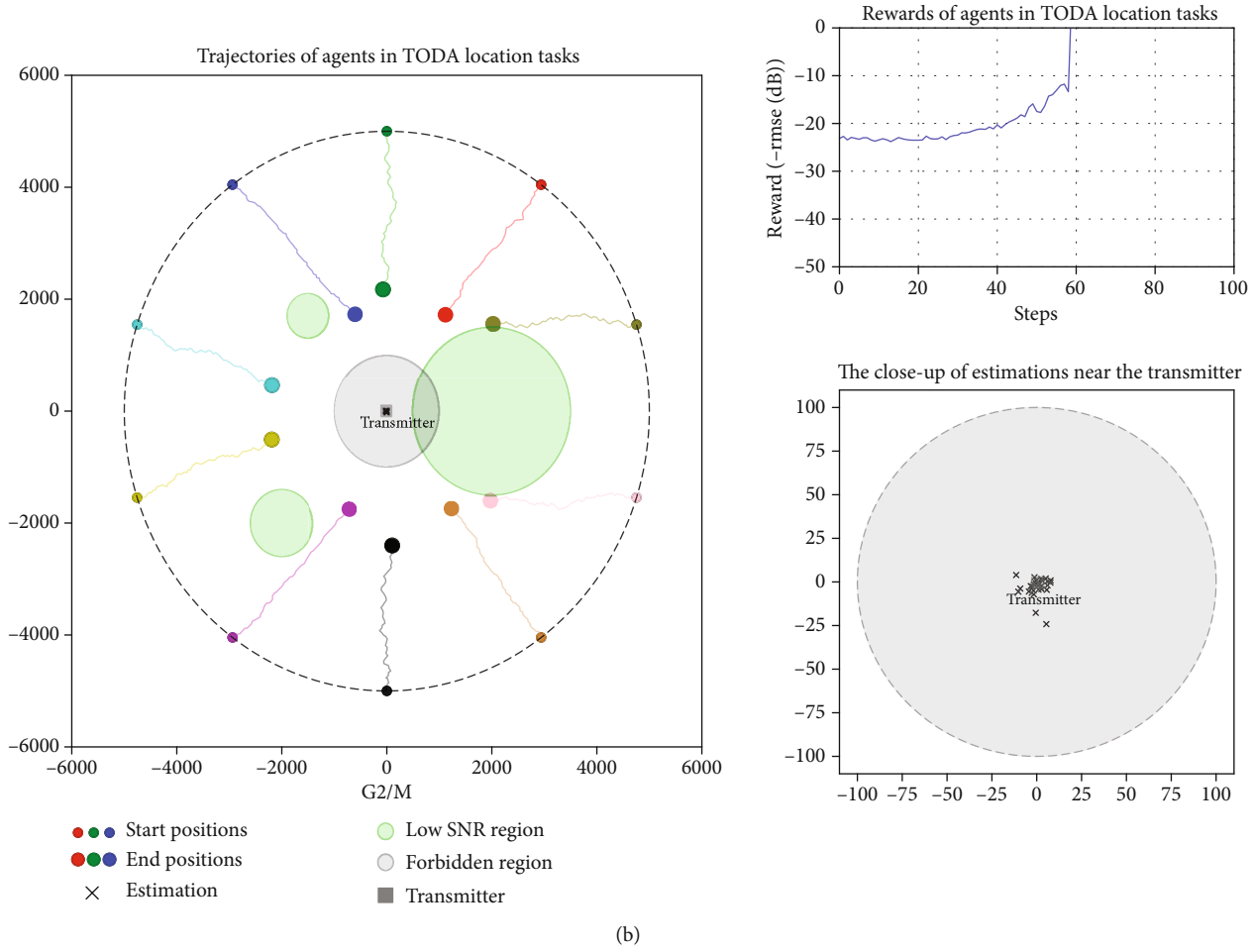
6.2. Setup. We model the passive location task as a multiagent decision-making problem, on which we evaluate our proposed method. According to the key components of reinforcement learning, the observation, action, and reward function are defined as follows.

6.2.1. Observations. The observation of agent i comprises the features of the received signals $f(z^i)$ and its position (x^i, y^i). The state of agent i is presented by $s^i = [f(z^i), x^i, y^i]$, where $f(\cdot)$ is a function that extracts features from the signals. In the experiment, the features of received signals refer to the SNRs.



(a)

Figure 4: Continued.



(b)

FIGURE 4: How trained agents accomplish passive location tasks collaboratively. (a) Number of agents $N = 7$. (b) Number of agents $N = 10$.

6.2.2. Action. The action of agent i is the position adjustment of itself, $[\Delta x^i, \Delta y^i]$. Hence, the agent will move to $[x^i + \Delta x^i, y^i + \Delta y^i]$ in the next time step. In the experiment, actions are clipped within an interval of $[-50, 50]$.

6.2.3. Reward. All the agents share the same reward in each time step. The reward function reflects the positioning precision, which is defined by

$$r_{\mathrm{RMSE}}(p, p_{\hat{a}}) = -10 \log_{10}\left(1 + \sqrt{\frac{1}{N_{\mathrm{est}}}\sum_{k=1}^{N_{\mathrm{est}}} \|p_{\hat{a}}^k - p_{\hat{a}}\|^2}\right), \quad (30)$$

where $p_{\hat{a}}$ is the position of the transmitter, $p_{\hat{a}}^k$ is the kth estimation of $p_{\hat{a}}$, and N_{est} denotes the estimation time in each time step. In the experiment, we have $N_{\mathrm{est}} = 100$.

For each agent, both the actor and the critic are designed as fully connected neural networks, which have two hidden layers of 256 neural units, and each layer is followed by an activation layer of *tanh* function. The actor maps the observation into actions. Specifically, it takes in the observation, then generates a two-dimensional Gaussian distribution, from which the action is sampled. The structure of the critic is similar to that of the actor, but the output is the value function that is used to optimize the actor according to the policy gradient theory.

The diffusion processes are completed one by one among agents but in a random order, and each agent updates its global state estimation from neighbors only once at each time step. Agents within 2 km are called neighbors. As shown in Figure 1, agent 2 is a neighbor of agent 1. Under the setting of Figure 1, agent 1 is able to access the observation and last action of agent 2 but cannot obtain these information of agent 3.

6.3. Ablation Study. To understand the influence of the agent number on the performance, we performed a series of experiments with different agent numbers ($N = 3, 5, 7, 10$) on the passive location task described above. Concretely, we focus on these indices that reflect the process of training across all the agents: averaged episode return, averaged episode length, policy entropy, and averaged number of neighbors. The maximum number of steps in an epoch is 800, with each episode executing no more than 100 steps.

Figure 2 shows the training curves after 1000 epochs across 5 random seeds (0,100,200,300,400,500) that initialize the agents. The optimizer and learning rate are Adam and 1×10^{-4}, respectively. Averaged episode return is a key indicator that reflects the ability of agents to accomplish the

given task. From Figure 2(a), it can be seen that when $N = 3$, agents cannot master useful skills to obtain a higher episode return. But the performance is improved with the increase of the agent number ($N = 5, 7, 10$). It is consistent with Figure 2(b), in which the averaged number of neighbors increases as the scale of agents becomes larger. The averaged number of neighbors reflects the connectivity of agents, which determines the diffusion efficiency of messages between agents. When agent number $N = 3$, agents may be scattered with distances out of the ability to exchange information. In that case, our message diffusion-based method reduced to independent agents, not being able to handle the cooperative task in a partial observation environment.

Figures 2(c) and 2(d) demonstrate the trend of mean policy entropy as well as the averaged episode length over agents across 1000 epochs of training. Both these two indicators drop as more agents take part in the task. The decrease of episode length indicates that agents can accomplish the task with fewer steps and find more elegant paths to optimal geometries collaboratively. The decline of policy entropy suggests that the agents become more confident in decision-making.

6.4. Comparison Study. In the comparison study, we investigate the advantages of our proposed method with two kinds of algorithms that are popular in the field of decentralized multiagent reinforcement learning. One is independent agents developed by algorithms such as IQL [16]. But in the passive location task, the action space is continuous, and we have every agent learning with its own actor-critic structure for handling partial observation, called *independent actor-critic* (IAC). Another similar decentralized algorithm that we make a comparison to is proposed in [20], where the agents update their neural networks by directly combining the parameters of neighbors. We refer to this method by *weight sharing* (WS).

The proposed method in this paper facilitates the collaboration of agents by message diffusion, which makes individual estimation of the global state more accurate. We performed our method (*Diffu* for short) and the contrast methods (IAC and WS) with different numbers of agents ($N = 3, 5, 7, 10$), and the results are demonstrated in Figure 3. In general, our method has superior performance in the experiments. It can be seen that for the proposed method, the more agents there are, the more advantages can be obtained. IAC failed in all the experiments due to the inherent challenge faced with a multiagent environment; i.e., from the perspective of an individual agent, the actions of other agents become nonstationary. WS attempts to address the challenge by introducing weight sharing among agents. It obtained the information of neighbors in an indirect way that combines neural network parameters of neighbors, but it is not so effective in the experiments. In the task of passive location, with more agents, message diffusion becomes easier so that each agent has a more accurate estimation of the global state, which is helpful to tackle the nonstationary problem.

The learned agents are able to perform passive location tasks effectively. Figure 4 shows the trajectories that navigate to an optimal geometry with different numbers of agents ($N = 7, 10$). In these two scenarios, agents can adjust the geometry collaboratively to improve the positioning precision and even master the skills such as taking a detour to avoid low forbidden regions or sacrifice immediate reward for a better geometry configuration.

7. Conclusion

This paper investigated the multiagent problem in a time-varying network, where agents only observe partial information and exchange information with neighbors. A fully decentralized actor-critic multiagent reinforcement learning algorithm based on message diffusion is proposed. Each agent is trained to make decisions depending on its own local observations and messages received from neighbors. This completely noncentral training and execution method overcomes the data collection challenges, especially when both the state space and the action space are massive. The convergence of the proposed algorithm with linear function approximation is guaranteed. Experimental results confirmed the convergence and effectiveness of the proposed algorithm. This decentralized method can be used in many other areas, such as packet routing in computer/wireless communications. In future work, more general function approximations will be employed to analyze the convergence of the algorithm.

Appendix

A. Proof of Theorem 4

Theorem A.1. *Under the conditions of Definition 1, Assumption 2, and Assumption 3, the gradient of the global objective function to agent i's policy is computed by*

$$\nabla_{\theta^i} J(\theta) = \mathbb{E}_{s\sim d_\theta(s), a\sim\pi_\theta(s,\cdot)}\left[\psi^i N^{-1}\sum_{j\in\mathcal{V}} A^j(s_{t,i}, a_{t,i})\right], \tag{A.1}$$

where $\psi^i = \nabla_{\theta^i}\log\pi_{\theta^i}(s_{t,i}, a_{t,i})$ *is the score function, and* A^j *is the local advantage function:*

$$A^i(s_{t,i}, a_{t,i}) = Q^i\left(s_{t,i}, a_{t,i}; \omega_t^i\right) - \sum_{a'\in\mathcal{A}^i}\pi_{\theta^i}\left(s^i, a', a_{t,-i}\right)Q^i \cdot \left(s^i, a', a_{t,-i}; \omega_t^i\right). \tag{A.2}$$

Proof. The gradient of global value function $V(s)$ is

$$\begin{aligned}\nabla_{\theta^i} V(s) &= \sum_{a\in\mathcal{A}}\left[\nabla_{\theta^i}\pi(s,a)Q(s,a) + \pi(s,a)\nabla_{\theta^i}Q(s,a)\right]\\ &= \sum_{a\in\mathcal{A}}\left[\nabla_{\theta^i}\pi(s,a)Q(s,a) + \pi(s,a)\nabla_{\theta^i}\left[\bar{r} - J(\theta) + \sum_{s'\in\mathcal{S}} p\left(s'\mid s,a\right)V\left(s'\right)\right]\right]\\ &= \sum_{a\in\mathcal{A}}\left[\nabla_{\theta^i}\pi(s,a)Q(s,a) + \pi(s,a)\left[-\nabla_{\theta^i}J(\theta) + \sum_{s'\in\mathcal{S}} p\left(s'\mid s,a\right)\nabla_{\theta^i}V\left(s'\right)\right]\right].\end{aligned} \tag{A.3}$$

Then, the gradient of the global objective is

$$\nabla_{\theta^i} J(\theta) = \sum_{a\in\mathscr{A}} \left[\nabla_{\theta^i} \pi(s,a) Q(s,a) + \pi(s,a) \sum_{s'\in\mathscr{S}} p\left(s' \mid s,a\right) \nabla_{\theta^i} V\left(s'\right) \right] - \nabla_{\theta^i} V(s). \tag{A.4}$$

Take summation for $d_\theta(s)$ on both sides of (A.4), and we have

$$\begin{aligned} \sum_{s\in\mathscr{S}} d_\theta(s) \nabla_{\theta^i} J(\theta) &= \sum_{s\in\mathscr{S}} d_\theta(s) \sum_{a\in\mathscr{A}} \nabla_{\theta^i} \pi(s,a) Q(s,a) \\ &\quad + \sum_{s\in\mathscr{S}} d_\theta(s) \sum_{a\in\mathscr{A}} \sum_{s'\in\mathscr{S}} p\left(s' \mid s,a\right) \nabla_{\theta^i} V\left(s'\right) \\ &\quad - \sum_{s\in\mathscr{S}} d_\theta(s) \nabla_{\theta^i} V(s). \end{aligned} \tag{A.5}$$

Notice that s is in a stable state; then,

$$\begin{aligned} \sum_{s\in\mathscr{S}} d_\theta(s) \nabla_{\theta^i} J(\theta) &= \sum_{s\in\mathscr{S}} d_\theta(s) \sum_{a\in\mathscr{A}} \nabla_{\theta^i} \pi(s,a) Q(s,a) \\ &\quad + \sum_{s'\in\mathscr{S}} d_\theta\left(s'\right) \nabla_{\theta^i} V\left(s'\right) - \sum_{s\in\mathscr{S}} d_\theta(s) \nabla_{\theta^i} V(s). \end{aligned} \tag{A.6}$$

Hence,

$$\nabla_{\theta^i} J(\theta) = \sum_{s\in\mathscr{S}} d_\theta(s) \sum_{a\in\mathscr{A}} \nabla_{\theta^i} \pi(s,a) Q(s,a). \tag{A.7}$$

Based on Assumption 2 in the paper, we have

$$\nabla_{\theta^i} J(\theta) = \sum_{s\in\mathscr{S}} d_\theta(s) \sum_{a\in\mathscr{A}} \nabla_{\theta^i} \pi(s,a) \frac{1}{N} \sum_{j\in\mathscr{V}} Q^j\left(s_j, a_j\right), \tag{A.8}$$

and also

$$\sum_{a\in\mathscr{A}} \nabla_{\theta^i} \pi(s,a) V(s_i) = V(s_i) \nabla_{\theta^i} \left[\sum_{a\in\mathscr{A}} \pi(s,a) \right] = 0. \tag{A.9}$$

Hence,

$$\begin{aligned} \nabla_{\theta^i} J(\theta) &= \sum_{s\in\mathscr{S}} d_\theta(s) \sum_{a\in\mathscr{A}} \nabla_{\theta^i} \pi(s,a) \frac{1}{N} \sum_{j\in\mathscr{V}} Q^j\left(s_j, a_j\right) \\ &= \sum_{s\in\mathscr{S}} d_\theta(s) \sum_{a\in\mathscr{A}} \nabla_{\theta^i} \pi(s,a) \frac{1}{N} \sum_{j\in\mathscr{V}} \left[Q^j\left(s_j, a_j\right) - V\left(s_j\right) \right] \\ &= \mathbb{E}_{s\sim d_\theta(s), a\sim\pi_\theta(s,\cdot)} \left[\nabla_{\theta^i} \log \pi_{\theta^i}(s_i, a_i) \frac{1}{N} \sum_{j\in\mathscr{V}} A^j\left(s_j, a_j\right) \right] \\ &= \mathbb{E}_{s\sim d_\theta(s), a\sim\pi_\theta(s,\cdot)} \left[\psi^i \frac{1}{N} \sum_{j\in\mathscr{V}} A^j\left(s_j, a_j\right) \right]. \end{aligned} \tag{A.10}$$

□

B. Proof of Convergence of Estimated Reward Function

Theorem B.1. *$\{\mu_t^i\}$ is bounded, namely, $\sup_t |\mu_t^i| < \infty$, $\forall i \in \mathscr{V}$.*

Proof. The update process of the estimated value of the reward function can be written as

$$\mu_{t+1}^i = \left(1 - \beta_{\omega,t}\right) \cdot \mu_t^i + \beta_{\omega,t} \cdot r_{t+1}^i, \tag{B.1}$$

and the asymptotics of which can be described by the following ordinary differential equation:

$$\dot{\mu}^i = -\mu^i + \sum_{s\in\mathscr{S}} d_\theta(s) \sum_{a\in\mathscr{A}} \pi_{\theta^i}(s,a) R^i(s,a). \tag{B.2}$$

Using $f(\mu^i)$ to represent the right side of (B.2), it is obvious that $f(\mu^i)$ is continuous to μ^i. Let $f_c(\mu^i) = f(c\mu^i) \cdot c^{-1}$, since $d_\theta(s)$ and $\pi^i(s,a)$ are probability functions, and the reward $R^i(s,a)$ is bounded, and then we have $f_\infty(\mu^i) = \lim_{c\longrightarrow\infty} = -\mu^i$. Therefore, the differential equation $\dot{\mu}^i = f_\infty(\mu^i)$ has a unique asymptotically stable equilibrium point. Meanwhile, since r_{t+1}^i is consistently bounded, there exists $K_0 \in \mathbb{R}$ which makes

$$\mathbb{E}\left[\left| r_{t+1}^i - \mathbb{E}\left(r_{t+1}^i\right) \right|^2 \right] \le K_0 \cdot \left(1 + \left|\mu_t^i\right|^2\right). \tag{B.3}$$

According to the stochastic approximation theory in Appendix C, μ_t^i converges. □

C. Stochastic Approximation Theory

Consider a stochastic approximation process for an n-dimensional random variable

$$x_{t+1} = x_t + \gamma_t [h(x_t, Y_t) + M_{t+1} + \beta_{t+1}], \quad t \ge 0, \tag{C.1}$$

where $\gamma_t > 0$ is the iteration step size, and $\{Y_t\}_{t\ge 0}$ is a finite Markov chain.

Assumption C.1. For the above random approximation process, the following assumption is made:

(1) $h : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is Lipschitz continuous for its first parameter

(2) $\{Y_t\}_{t\ge 0}$ is an irreducible Markov chain, of which the stationary distribution is π

(3) The step size satisfies $\gamma_t > 0$, $\sum_t \gamma_t = \infty$, and $\sum_t \gamma_t^2 < \infty$

(4) $\{M_t\}$ is a martingale difference sequence, namely, $\mathbb{E}[M_{t+1} \mid x_\tau, M_\tau, Y_\tau, \tau \le 0] = 0$, and it satisfies

$$\mathbb{E}\left[\|M_{t+1}\|^2 \mid x_\tau, M_\tau, Y_\tau, \tau \le 0\right] \le K \cdot \left(1+\|x_t\|^2\right). \quad \text{(C.2)}$$

(5) Sequence $\{\beta_t\}$ is a bounded random sequence, and $\beta_t \longrightarrow 0, t \longrightarrow 0$

The iteration of stochastic approximation is presented in (C.1), and its asymptotic property can be captured by the following differential equation:

$$\dot{x} = \bar{h}(x) = \sum_i \pi(i) h(x, i). \quad \text{(C.3)}$$

Assuming (C.3) has a global asymptotically stable equilibrium point, the following two theorems hold.

Theorem C.2. *Based on Assumption C.1 (1)–(4), if* $\sup_t \|x_t\| < \infty$, $x_t \longrightarrow x^*$.

Theorem C.3. *Based on Assumption C.1 (1)–(4), the limit of* $\lim_{c \longrightarrow \infty} \bar{h}(cx)/c = h_\infty(x)$ *exists. If the differential equation* $\dot{y} = h_\infty(y)$ *has origin as the unique globally asymptotically stable equilibrium, then* $\sup_t \|x_t\| < \infty$.

Data Availability

The data (experiment environment and source code) is developed by our research team, and we will open the source code at an appropriate time. Requests for access to these data should be made to Shengxiang Li, lishengxiangzz@163.com.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

[3] Y. Yang, B. Kiumarsi, H. Modares, and C. Xu, "Model-free _-policy iteration for discrete-time linear quadratic regulation," in *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[4] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[5] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," in *Proceedings of 4th International Conference on Learning Representations*, 2016.

[7] L. Busoniu, R. Babuska, and B. De Schutter, "Multi-agent reinforcement learning: a survey," in *Proceedings of 9th International Conference on Control, Automation, Robotics and Vision*, pp. 1–6, Singapore, 2006.

[8] T. Nguyen, B. Xue, P. Andreae, and M. Zhang, "A hybrid evolutionary computation approach to inducing transfer classifiers for domain adaptation," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[9] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A very condensed survey and critique of multiagent deep reinforcement learning," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2146–2148, 2020.

[10] O. Vinyals, I. Babuschkin, W. M. Czarnecki et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[11] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2018.

[12] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis, "Optimal and approximate Q-value functions for decentralized POMDPS," *The Journal of Artificial Intelligence Research*, vol. 32, pp. 289–353, 2008.

[13] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proceedings of 35th International Conference on Machine Learning*, pp. 6846–6859, 2018.

[14] P. Sunehag, G. Lever, A. Gruslys et al., "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 2018 International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 2085–2087, 2018.

[15] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proceedings of Conference on Neural Information Processing Systems*, pp. 6379–6390, 2017.

[16] M. Lauer and M. Riedmiller, "Distributed reinforcement learning in multi-agent networks," in *Proceedings of the 7th International Conference on Machine Learning*, pp. 296–299, St. Martin, France, 2000.

[17] M. Tan, "Multi-agent reinforcement learning: independent vs. cooperative agents," *Proceedings ofthe tenth international conference on machine learning*, pp. , 1993330–337, 1993.

[18] G. Laurent, L. Matignon, and N. le Fort-Piat, "The world of independent learners is not Markovian," *International Journal of Knowledge-based and Intelligent Engineering Systems*, vol. 15, no. 1, pp. 55–64, 2011.

[19] S. Kar, J. M. F. Moura, and H. V. Poor, "$\mathcal{QD}$-learning: a collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, 2013.

[20] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *35th International Conference on Machine Learning*, pp. 9340–9371, 2018.

[21] S. Y. Tu and A. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, 2012.

[22] R. S. Sutton, M. A. DA, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function

approximation," *Advances in Neural Information Processing Systems*, pp. , 20001057–1063, 2000.

[23] S. M. Ross, *Stochastic Processes*, China Machine Press, 2nd edition, 2013.

[24] T. Degris, M. White, and R. S. Sutton, "Off-policy actor-critic," in *ICML'12: Proceedings of the 29th International Conference on International Conference on Machine Learning*, pp. 179–186, 2012.

[25] H. Prasad, L. A. Prashanth, and S. Bhatnagar, "Two-timescale algorithms for learning Nash equilibria in general-sum stochastic games," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1371–1379, 2015.

[26] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*, Hindustan Book Agency (lndia), 2008.

[27] S. Li, G. Liu, S. Ding, H. Li, and O. Li, "Finding an optimal geometric configuration for TDOA location systems with reinforcement learning," *IEEE Access*, vol. 9, pp. 63388–63397, 2021.

[28] H. J. Kushner and D. Clark, "Stochastic approximation methods for constrained and unconstrained systems," Springer Science & Business Media, 1978.

[29] H. J. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, Springer-Verlag, 2nd edition, 2003.