EasyChair Preprint

№ 11382

Reinforcement Learning-Based Consensus-Reaching in Large-Scale Social Networks

Shijun Guo, Haoran Xu, Guangqiang Xie, Di Wen, Yangru Huang and Peixi Peng

November 25, 2023

Reinforcement Learning-Based Consensus Reaching in Large-Scale Social Networks

Shijun Guo[1†][0009−0008−4556−4818], Haoran Xu[2,3†][0000−0002−9330−2475], Guangqiang Xie[1✉][0000−0002−9857−2140], Di Wen[2,3][0000−0003−2214−2088], Yangru Huang[4][0000−0002−7865−9388], and Peixi Peng[3,4][0000−0002−7427−8764]

[1] School of Computer Science and Technological, Guangdong University of Technology, No.100 Waihuanxi Road, HEMC, Guangzhou 510006, Guangdong, China
{3120005895, xiegq}@gdut.edu.cn

[2] Peng Cheng Laboratory, Xingke 1st Street, Shenzhen 518066, Guangdong, China
{xuhr, wend}@pcl.ac.cn

[3] School of Intelligent Systems Engineering, Sun Yat-sen University, No.66 Gongchang Road, Shenzhen 510275, Guangdong, China
{xuhr9, wend25}@mail2.sysu.edu.cn

[4] Department of Computer Science and Technology, Peking University, Beijing, 100871, China
yrhuang@stu.pku.edu.cn, pxpeng@pku.edu.cn

Abstract. Social networks in present-day industrial environments encompass a wide range of personal information that has significant research and application potential. One notable challenge in the domain of opinion dynamics of social networks is achieving convergence of opinions to a limited small number of clusters. In this context, designing the communication topology of the social network in a distributed manner is a particularly difficult. To address this problem, this paper proposes a novel perception model for agents. The proposed model, which is based on bidirectional recurrent neural networks, can adaptively reweight the influence of perceived neighbors in the convergence process of opinion dynamics. Additionally, effective differential reward functions are designed to optimize three objectives: convergence degree, connectivity, and cost of convergence. Lastly, a multi-agent exploration and exploitation algorithm based on policy gradient is designed to optimize the model. Based on the reward values in inter-agent interaction process, the agents can adaptively learn the neighbor reweighting strategy with multi-objective trade-off abilities. Extensive simulations demonstrate that the proposed method can effectively reconcile conflicting opinions among agents and accelerate convergence.

Keywords: Social network · Opinion dynamics · Reweighting perception · Reinforcement learning.

†: indicates equal contribution.
✉: corresponding author

1 Introduction

Recently, social network group decision-making (SNGDM) has attracted increasing attention as a valuable tool for understanding and explaining human actions [9, 14]. SNGDM frameworks have excellent research and application potential in many fields, such as supplier selection [1], public opinion management [31, 11], political elections [3, 34], markets [16, 2] and transportation [12]. To select the best alternative from a set of potential candidates, SNGDM involves a set of individuals, known as agents, who can express their opinions and communicate with their neighbors. Opinions and beliefs are crucial factors that influence our behavior, thereby defining our individuality and driving our actions [13, 23, 10, 8]. Opinion evolution, also known as opinion dynamics, represents the process of modification of an agent's opinions by merging them with those of other agents, resulting in the formation of a stable structure. This process may involve consensus, polarization, or fragmentation [10].

A notable challenge in SNGDM is the achievement of a general and unanimous agreement among all agents [22, 5]. With the rapid development of wireless communication networks and Internet-based technologies, we can now exchange opinions with a large number of people in real-time. The large-scale consensus reaching Process (LSCRP) in agents' opinions in SNGDM takes into account the opinions of agents throughout a social network [10]. The network represents the interaction rule between agents and plays a critical role in opinion dynamics [15, 19]. A growing body of literature has recognized the importance of network topology in the fusion rules of opinion dynamics in the LSCRP. Lu *et al.* [17] allowed agents to express their trust values and relationship strengths with other agents to better reflect the actual social network and improve the efficiency of LSCRP. Chao *et al.* [6] constructed a two-layer network topology to address incomplete social relationships among agents on a large-scale. This framework could reconcile conflicting preferences and accelerate LSCRP at a minimal cost. Additional work on LSCRP can be found in [30, 22]. Notably, the existing methods typically consider the peer-to-peer network topology, in which each agent communicates directly with all other perceived neighbors with the equal weight to update its own state.

Although consensus reaching processes have been widely studied and LSCRP investigations have achieved promising results, research on LSCRP within the context of SNGDM is still in its nascent stages [9]. The requirement of a high consensus level and presence of incomplete social relationships may render communication and opinion evolution among agents complex and challenging, especially in a large-scale scenario [6]. However, existing LSCRP methods consider only one-hop-based connections, in which agents interact with similar agents based on specific contexts, and ignore the more efficient interaction patterns that can facilitate consensus. These problems can be addressed by introducing multi-agent reinforcement learning (MARL), which has emerged as a powerful technology for accomplishing dynamic tasks online [32]. For example, Zhang *et al.* [33] used mean-field theory to decompose the joint action from the individual perspective in cooperative settings. Moreover, Sun *et al.* [21] solved the real-time

Volt-Var control problem by using the multi-agent deep deterministic policy gradient method in a cooperative setting. However, the integration of MARL with LSCRP frameworks remains largely unexplored.

Considering these aspects, this study is focused on promoting consensus among a large group of people (i.e., in a multi-agent system) in a social network. We allow the agents to adaptively and discriminatively select the influence of locally perceived neighbors, thereby reconciling conflicting opinions and fostering unanimous consensus among agents at a higher rate. The main contributions of this study can be summarized as follows:

- We propose a novel agent perception model based on bidirectional long short-term Memory (LSTM), which adaptively reweights the influence of perceived local neighbors.
- We devise three types of differential reward functions within social networks to facilitate reinforcement learning.
- We design a multi-agent exploration and exploitation algorithm based on policy gradient to effectively train the agent perception model.

2 Model Formulation

This section describes the proposed model and reward functions. Section 2.1 introduces the decision-making model that adaptively adjusts the weights of neighboring states. Section 2.2 outlines the three goal-oriented differential reward functions designed to facilitate the learning process of agents.

2.1 Model Settings

The number of neighbors perceived by each agent is uncertain during the evolution of opinions, and each local neighbor needs to be evaluated in the decision-making process. Therefore, we use a recurrent neural network (RNN), which is a suitable model for time-series analysis, to address the problem of uncertain local perception input and uncertain decision length. Given that the agents must consider the overall context information of all perceived neighbors to make judgments, the perceived neighbors are encoded by extending the conventional bidirectional RNN.

As shown in Fig. 1, we encode the perceived neighbors based on the bidirectional LSTM [29] model, which is a widely used variant of RNN. The input to the proposed model is the set of state values $s_i(k) = \{x_j(k) : j \in N_i(k)\}$ of all perceived neighbors of agent i and their corresponding index $j : j \in N_i(k)$ at time k. After passing through the bidirectional LSTM network, the output is the action $a_i(k)$ for the opinion value of each neighbor, which represents the reweighting probability of each neighbor. Thus, the output layer of the LSTM model contains the SoftMax activation function, which ensures that the sum of the reweighting probabilities is 1. Therefore, the dimension of action $a_i(k)$ is consistent with that of the input state set, i.e.,$[N_i(k), 1]$. Additionally, to endow

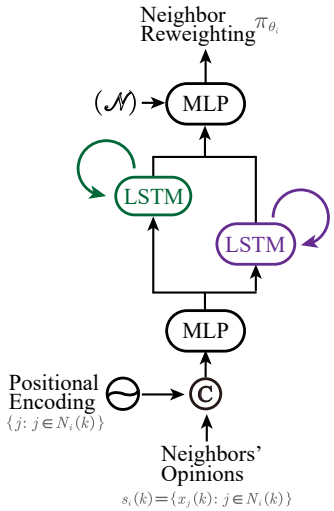


Fig. 1: Adaptively reweighting model based on LSTM

the agents with exploration abilities to improve the robustness of the learning process, we add the noise $p \times \mathscr{N}$ to the output layer weight of the final action, where $\mathscr{N}$ follows the standard normal distribution, and p is a temperature parameter that controls the scale of noise $\mathscr{N}$.

The output action $a_i(k) \sim \pi_{\theta_i}$ of the model represents the aggregation of the influence weight $\pi^j_{\theta_i} \in (0, 1)$ of each neighbor. The weights of the non-neighbors of agent i are set as zero. To obtain the new communication topology matrix $L(k)$ after processing through the proposed model, we aggregate the weights of the neighbors and non-neighbors of agent i in the order of agent index. Specifically, $L(k) = [l_{ij}(k)]$, where each element $l_{ij}(k)$ denotes the new communication topology relationship within the network, as defined in Eq. (1).

$$l_{ij} = \begin{cases} \pi^j_{\theta_i}, & j \in N_i(k) \\ 0, & j \notin N_i(k) \end{cases} \tag{1}$$

2.2 Reward Settings

The design of the reward function significantly affects the learning process of agents [20]. Furthermore, the reward function must be formulated considering the balance between the index coefficient and learnability. Therefore, we design the folllowing differential reward function for the social network:

$$r_i(k) = \alpha \mathscr{G}_1(X(k+1)) + \beta \mathscr{G}_2(X(k+1)) + \gamma \mathscr{G}_3 \tag{2}$$

where $\mathscr{G}_1(\cdot)$, $\mathscr{G}_2(\cdot)$ and $\mathscr{G}_3$ represent the three objectives, with α, β, and γ representing their temperature coefficients respectively.

$\mathscr{G}_{\mathbf{1}}(\cdot)$ enhances the capability of the agents in enhancing the consensus degree of the system. The convergence degree is quantified considering the standard deviation of opinion values in the multi-agent system:

$$\mathscr{G}_1 = g_1\left(X\left(k+1\right)\right) - g_1\left(X_{-i}\left(k+1\right)\right) \tag{3}$$

$$g_1\left(X\left(k+1\right)\right) = \frac{std\left(X\left(k+1\right)\right)}{n} \tag{4}$$

where $n = |V|$ represents the number of agents in the system, $X_{-i}(k+1)$ represents the state after the exclusion of agent i's state from the global state, and $std(\cdot)$ represents the standard deviation operation. The range of g_1 depends on the initial range of the system state and is typically $\left[0, \frac{|O_{max}-O_{min}|}{n}\right)$, where O_{max} and O_{min} represent the maximum and minimum values of the system state, respectively. In the Hegselmann-Krause model, (HK) [4], O_{max} is typically less than 10. A g_1 value closer to 0 corresponds to superior convergence performance of the system. The range of $\mathscr{G}_1$ is $[-\frac{|O_{max}-O_{min}|}{n}, \frac{|O_{max}-O_{min}|}{n}]$, and a value closer to $-\frac{|O_{max}-O_{min}|}{n}$ indicates a lower divergence degree of agent i with respect to the complete system.

$\mathscr{G}_{\mathbf{2}}(\cdot)$ enables agents to improve the connectivity density of the system. The network topology represents the communication pattern underlying opinion dynamics and plays a key role in convergence theory [18]. Therefore, we quantify the connectivity by the density of the network topology:

$$\mathscr{G}_2 = g_2\left(X\left(k+1\right)\right) - g_2\left(X_{-i}\left(k+1\right)\right) \tag{5}$$

$$g_2\left(X\left(k+1\right)\right) = \frac{\sum_{i\in V}\left|N_i\left(k+1\right)\right|}{n^2} \tag{6}$$

where $X_{-i}(k+1)$ has the same meaning as that in Eq. (3), and $\sum_{i\in V}\left|N_i\left(k+1\right)\right|$ represents the number of connections of the system at time k. Thus, the range of g_2 is $[0, 1]$, and a value is closer to 1 indicates a higher degree of connectivity in the system. The range of $\mathscr{G}_2$ is $[0, 1)$, and a value closer to 1 indicates a greater influence of agent i on the density of the system distribution.

$\mathscr{G}_{\mathbf{3}}$ enables agents to reduce the number of steps required to achieve a consensus. Therefore, we intuitively introduce the penalty for each convergence step:

$$\mathscr{G}_3 = -0.01 \tag{7}$$

$\mathscr{G}_3$ accumulates with the number of steps. During a round of evolution, $\mathscr{G}_3$ takes a value in $[0, -0.01 \times \hbar]$, where $\hbar$ represents the number of steps required to achieve stability. A cumulative value of $\mathscr{G}_3$ closer to 0 indicates that fewer steps are required to achieve stability.

3 Algorithm

The policy gradient-based reinforcement learning algorithm parameterizes the agent strategy and then directly optimizes it by maximizing the expected cumulative return [25]. This method can effectively enable iterative optimization

Algorithm 1: Policy-gradient-based exploration and exploitation algorithm

Input: maximum training episode M, maximum time step T, learning batch size B

Output: the parameter θ of each agent's adaptively reweighting policy network

1 Initialize the policy parameter θ, the experience replay-buffer pool D, and the weights α, β, γ of differential reward function;
2 **for** e=1 **to** $|M|$ **do**
3 Reset and initialize the environment to obtain the global initial state $X(k)$ of the system;
4 Receive the temperature p to control the noise;
5 **for** k=1 **to** $|T|$ **do**
6 Each agent perceives local neighbors' state $s_i(k)$ according to Eq. (8);
7 Each agent reweights neighbors based on $a_i(k)$, and obtains $L_i(k)$ ad described in Sec. 2.1;
8 Each agent receives an instant differential reward according to Eq. (2);
9 The environment with states $X(k)$ evolves to $X(k+1)$ according to selected neighbors for all agents and Eq. (10);
10 Store the experience samples $(s_i(k), a_i(k), r_i(k))$ of all agents in the experience replay-buffer pool D;
11 **if** *done* **then**
12 `Break`;
13 **end**
14 **end**
15 Randomly and uniformly select trajectory samples of agents with batch size B from the experience pool D;
16 Calculate the gradient $\nabla_\theta U(\theta)$ via Eqs. (14) and (15);
17 Update policy parameters θ via Eq. (17);
18 **end**
19 **return** θ

during agent exploration and address the challenges associated with the continuous action space. Based on the modeling and analysis of the state, action, and reward functions, as discussed in the previous sections, this section describes the process flow of exploration and exploitation, outlined in Algorithm 1.

For ease of reference in the following derivations, we use τ_i to denote the continuous state-action pairs $(s_i(0), a_i(0), \cdots, s_i(H), a_i(H))$ of agent i, generated through its interactions with the environment, where H denotes the length of the sequence. The proposed algorithm follows the distributed testing and centralized training framework and consists of two parts, i.e., the multi-agent exploration stage and strategy exploitation and updating stage. These stages are explained in the following sections.

Multi-agent exploration stage (lines 3-13 in Algorithm 1). In the process of exploration, each agent obtains the state value $s_i(k)$ of its neighbors within

its perception range, as indicated in Eq. (8).

$$N_i(k) = \{j \in V : |x_i(k) - x_j(k)| < 1\} \tag{8}$$

Then, each agent calculates the weight of its neighbors based on the proposed model, as shown in Fig. 1. We obtain a new communication topology matrix $L(k)$ using Eq. (1). The opinions of all agents evolve synchronously as

$$x_i(k+1) = \sum_{j \in N_i(k)} x_j(k) \times l_{ij}(k) \tag{9}$$

Each agent $V_i \in V$ in the system maintains an opinion $x_i(k)$, represented by a real number, for a given issue. Let $X(k) = [x_1(k), x_2(k), \cdots, x_n(k)]^T$ be the opinion profile of all agents at time k. In this case, the model (9) can be rewritten in the following matrix form:

$$X(k+1) = L(k)X(k) \tag{10}$$

To ensure that the agents can automatically and adaptively select the exploration method according to episode, the noisy weight parameter p is decreased as the number of episodes increases.

Subsequently, to learn the policy parameters in the multi-agent exploitation stage, each agent stores the local perception state $s_i(k)$, local action $a_i(k)$ and immediate reward $r_i(k)$ at each time step in the experience buffer pool in chronological order based on the longest time span of exploration. At the end of each episode, the sequences of agents are selected for learning through uniform random sampling. If the system reaches a state of consistency, i.e, if all agents converge to a cluster, the parameter $done = true$ indicates early termination of the current round of exploration (lines $11 - 13$).

Strategy exploitation and updating stage (lines 15-17 in Algorithm 1). At the end of each episode, the experience sequences of agents are selected for learning through uniform random sampling. To facilitate the following derivation, we use τ_i to denote the continuous state-action pairs $(s_i(0), a_i(0), \cdots, s_i(H), a_i(H))$ of agent i, where H denotes the length of the sequence. For updating the policy parameters, agents constantly explore and exploit their policies to maximize the expected cumulative return in the future:

$$U(\theta) = \mathbb{E}_{\pi_{\theta_i}}\left[\sum_k \left(r_i(k) \left| s_i(k), a_i(k)\right.\right)\right] \approx \sum_{\tau_i} P(\tau_i|\theta_i) R_i(\tau_i) \tag{11}$$

$$R_i(\tau_i) = \sum_{k=0}^{H} r_i(k) \tag{12}$$

We use the gradient descent method to find the gradient $\nabla_\theta U(\theta)$ of objection function $U(\theta)$:

$$\begin{aligned}\nabla_\theta U(\theta) &= \nabla_\theta \sum_{\tau_i} P(\tau_i|\theta_i) R_i(\tau_i) \\ &= \sum_{\tau_i} P(\tau_i|\theta_i) \frac{\nabla_\theta P(\tau_i|\theta_i)}{P(\tau_i|\theta_i)} R_i(\tau_i) \\ &= \sum_{\tau_i} P(\tau_i|\theta_i) \nabla_\theta \log P(\tau_i|\theta_i) R_i(\tau_i)\end{aligned} \tag{13}$$

According to Eq. (13), the gradient of $U(\theta)$ contains $P(\tau_i|\theta_i)$ and $\nabla_\theta \log P(\tau_i|\theta_i) R_i(\tau_i)$. Because $P(\tau_i|\theta_i)$ represents the probability of occurrence of trajectory τ_i, the gradient can be equivalent to the expectation of $\nabla_\theta \log P(\tau_i|\theta_i) R_i(\tau_i)$. Therefore, we estimate the gradient through average approximation based on the experience of the sampled trajectories:

$$\begin{aligned}\nabla_\theta U(\theta) &= \sum_{\tau_i} P(\tau_i|\theta_i) \nabla_\theta \log P(\tau_i|\theta_i) R_i(\tau_i) \\ &\approx \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau_i|\theta_i) R_i(\tau_i)\end{aligned} \tag{14}$$

Furthermore, the gradient calculated using Eq. (14) can be intuitively understood as follows: The algorithm increases and decreases the probability of occurrence of trajectories with high and low reward, respectively. Then, we solve the only uncertainty $\nabla_\theta \log P(\tau_i|\theta_i)$ in Eq. (14):

$$\begin{aligned}&\nabla_\theta \log P(\tau_i|\theta_i) \\ &= \nabla_\theta \log \left[\prod_{k=0}^{H} \begin{matrix} P(s_i(k+1)|s_i(k), a_i(k)) \times \\ \pi_{\theta_i}(a_i(k)|s_i(k)) \end{matrix} \right] \\ &= \nabla_\theta \left[\begin{matrix} \sum_{k=0}^{H} \log P(s_i(k+1)|s_i(k), a_i(k)) + \\ \sum_{k=0}^{H} \log \pi_{\theta_i}(a_i(k)|s_i(k)) \end{matrix} \right] \\ &= \nabla_\theta \left[\sum_{k=0}^{H} \log \pi_{\theta_i}(a_i(k)|s_i(k)) \right] \\ &= \sum_{k=0}^{H} \nabla_\theta \log \pi_{\theta_i}(a_i(k)|s_i(k))\end{aligned} \tag{15}$$

In Eq. (15), $P(s_i(k+1)|s_i(k), a_i(k))$ represents the system dynamics. Because the dynamics do not include the policy parameter θ, it can be deleted. Subsequently, we obtain the final policy gradient as:

$$\nabla_\theta U(\theta) \approx \frac{1}{m} \sum_{i=1}^{m} \sum_{k=0}^{H} \nabla_\theta \log \pi_{\theta_i}(a_i(k)|s_i(k)) r_i(k) \tag{16}$$

Table 1: Parameter settings in simulations

Param.	Explanation	Value
M	Maximum number of episodes	150
T	Maximum time step	20
B	Batch size	300
ζ	Learning rate	2e-3
n	Number of agents	20 to 100
ω	Density of agents	5 or 10
r_c	Perception range of agents	1
α, β, γ	Temperature coefficients of rewards	-1,1,1
$interval$	Initial state range	$[0, 4]$ or $[0, 10]$
th	Convergence threshold	1e-2
$sdim$	Input dimension of state in LSTM	2
$adim$	Output dimension of action in LSTM	1
$hdim$	Hidden dimension in LSTM	36
$hlays$	Number of hidden layers in LSTM	2

Table 2: Ablation study

Exp. ID	$\mathscr{G}_1$	$\mathscr{G}_2$	$\mathscr{G}_3$	Number of clusters	Convergence step
A1	✓	✗	✗	4.2 ± 0.39	11.5 ± 0.49
A2	✓	✗	✓	4.3 ± 0.45	9.0 ± 2.36
A3	✗	✓	✗	4.4 ± 0.48	11.2 ± 2.82
A4	✗	✓	✓	4.1 ± 0.51	9.9 ± 2.31
A5	✓	✓	✗	$\mathbf{3.7 \pm 0.45}$	9.6 ± 1.2
A6	✓	✓	✓	3.7 ± 0.78	$\mathbf{8.4 \pm 1.35}$

Lastly, the agent's policy parameters are updated through the steepest descent method with the learning rate ζ:

$$\theta \leftarrow \theta + \zeta \nabla_\theta U(\theta) \tag{17}$$

4 Experiment

We develop a simulation environment for opinion dynamics using Python 3.7.0 and model the agent policy-gradient network with an LSTM architecture using PyTorch 1.2.0. Table 1 lists the parameters used in the experiment. To comprehensively analyze the superiority and effectiveness of our method, we use the 'convergence step' and 'number of cluster' as the evaluation metrics. The convergence step refers to the number of steps required for the system to reach a state in which the distance between any agent and its neighbors is less than the threshold th'. The achievement of system stability with fewer convergence steps and a smaller number of clusters corresponds to a superior performance [26–28] The experimental results, demonstrate that the integration of MARL with LSCRP can help reconcile conflicting opinions and promote unanimous consensus among all agents.

4.1 Ablation Study

First, we verify the effectiveness of the proposed model using the differential reward function and compare the model performance under different reward

combinations, as presented in Table 2. The settings involve a system opinion range of $[0, 10]$ with $n = 100$ and $\omega = 10/r_c$. All experiments are conducted 10, and the mean and standard deviation are reported.

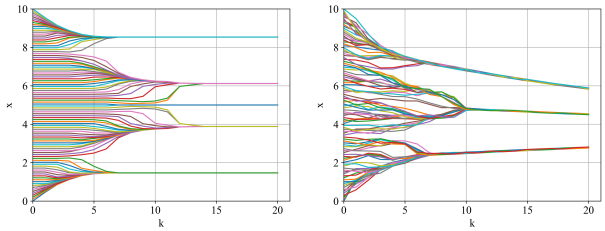
- **A1** vs. **A3** vs. **A5**: Compared with the scenarios in which only $\mathscr{G}_1$ (**A1**) or $\mathscr{G}_2$ (**A3**) is considered, the incorporation of both $\mathscr{G}_1$ and $\mathscr{G}_2$ (**A5**) leads to a 13.9% reduction in the average number of steps required for the system to reach stability and a 15.3% decrease in the average number of clusters at stability. Thus, using a combination of rewards $\mathscr{G}_1$ and $\mathscr{G}_2$ can help decrease the number of clusters and steps required to achieve system stability.
- **A1** vs. **A2**, and **A3** vs. **A4**: The convergence steps of **A2** and **A4** are 21.4% and 11.6% lower than those of **A1** and **A3**, respectively. This finding indicates that considering reward $\mathscr{G}_3$ allows agents to learn to reduce the number of steps required for system stability with an insignificant change in the number of clusters.
- **A5** vs. **A6**: By comprehensively considering rewards $\mathscr{G}_1$, $\mathscr{G}_2$ and $\mathscr{G}_3$, **A6** achieves a 12.5% reduction in convergence step compared with **A5**. This finding indicates that incorporating reward $\mathscr{G}_3$ in addition to $\mathscr{G}_1$ and $\mathscr{G}_2$ can further decrease the number of steps required to stabilize the system.

4.2 Comparative Analyses

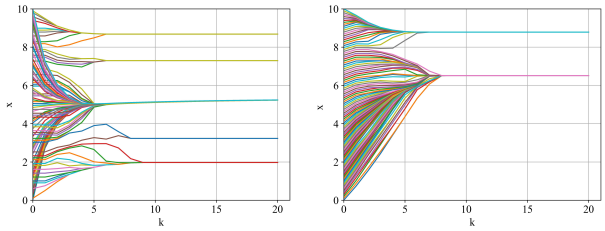
We access the effectiveness of the model in terms of the convergence step and number of clusters when the system achieves stability. The number of convergence clusters indicates the enhancement of consistency of the model, and the convergence step indicates the number of steps required by the model to achieve stability. We compare the proposed method with the HK, common-neighbor rule (CNR) [24], group-pressure (GP) methods [7]. Because the objective is to enhance consistency, we aim to ensure convergence to fewer clusters in fewer steps. The relevant parameters in the CNR and GP models are uniformly set as $\beta = 0$, $m = 1$ and $p_i = \lambda = 0.5$.

Existing LSCRP methods consider only one-hop-based connections, in which agents interact with similar agents, and ignore the efficient interaction patterns that can facilitate consensus. We illustrate the evolution process of the three baselines and our method with an initial range of $[0, 10]$ and $n = 100$. The following observations are made:

- HK evolves by obtaining the average value of neighbors, which makes it difficult to control the numbers of steps and clusters when the system is stable (Fig. 2a).
- CNR selects local and long-range neighbors according to the confidence bounds and a common-neighbor rule, respectively. Therefore, CNR requires a large amount time to reach a consensus (Fig. 2b).
- GP takes group pressure into consideration, leading to the formation of inner opinions within the agents' bounded confidence. However, the ambient pressure reduces communication between agents, resulting in more clusters (Fig. 2c).



(a) HK (b) CNR



(c) GP (d) ours

Fig. 2: Results of the proposed and baseline methods

Table 3: Statistics of comparison simulations

ID	Init. range	n	ω	Method	Cluster number	Convergence step
M1	$[0, 4]$	20	$\frac{5}{r_c}$	HK	2	5
				CNR	1	5
				GP	1	5
				ours	**1**	**5**
M2	$[0, 4]$	40	$\frac{10}{r_c}$	HK	1	9
				CNR	1	4
				GP	1	4.80 ± 0.75
				ours	**1**	$\mathbf{4.45 \pm 0.50}$
M3	$[0, 10]$	50	$\frac{5}{r_c}$	HK	5	11
				CNR	$\mathbf{3.33 \pm 0.94}$	19.8 ± 0.60
				GP	3.90 ± 0.70	14.90 ± 6.32
				ours	3.45 ± 0.50	$\mathbf{10.72 \pm 1.81}$
M4	$[0, 10]$	100	$\frac{10}{r_c}$	HK	5	14
				CNR	8.20 ± 2.40	20.00 ± 0.00
				GP	4.50 ± 0.50	18.00 ± 3.55
				ours	$\mathbf{2.70 \pm 0.45}$	$\mathbf{9.10 \pm 1.51}$

– The proposed method enable agents to adaptively learn the neighbor reweighting strategy with multi-objective trade-off ability. The agents required only a few steps to reach a consensus with a small number of clusters (Fig. 2d).

Table 3 summarizes the statistical results of the comprehensive comparisons. The experiments are conducted 10 times after 100 training runs, and the mean and standard deviation of the metrics are reported.

- **Horizontal analysis**: We analyze the performance of different algorithms under the same density. The proposed algorithm requires the fewest number of steps to achieve stability. With the improvement of the range (**M2**, **M4**), the advantage of our algorithm in terms of the number of steps is amplified.
- **Longitudinal analysis**: Under the same range, as the density increases, the numbers of steps and clusters associated with our algorithm decreases or remain stable. In high-range situations (**M3, M4**), the numbers of steps and clusters associated with the baselines increase as the density increases, whereas those of our algorithm steadily decrease.

Notably, in high situations (**M4**), the proposed method achieves an average reduction of 51% and 46.3% in the number of clusters and convergence step, respectively, compared with the baselines. These demonstrates the potential of the proposed method in resolving disagreements among agents and accelerating the consensus-building process.

5 Conclusions

With the objective of enhancing the consensus of opinion dynamics in the field of social networks, an intelligent perception model based on MARL is developed. For the convergence process, we first design an adaptive reweighting model based on bidirectional LSTM to capture the perception capability. Then, we formulate the corresponding differential reward function based on three types of goals in the opinion dynamics scenario. Finally, through the multi-agent exploration and strategy exploitation algorithm based on the policy gradient, the agents are allowed to adaptively learn an efficient neighbor reweighting strategy with multi-objective trade-off during their interaction. The experimental results verify that the proposed method can enable agents to adaptively reweight the influence of neighbors while exhibiting multi-objective trade-off abilities and effectively reconcile opinions with large differences in the social network system. Thus, the number of clusters at stability is reduced, and the convergence process is accelerated.

In future work, we will focus on the consistency enhancement method with attention mechanisms and privacy protection in social networks and verify the effectiveness and generalization ability of the proposed approach in real opinion dynamics scenarios.

Acknowledgements This work is partially supported by National Natural Science Foundation of China, Grant Nos. 62006047 and 618760439, Guangdong Natural Science Foundation, Grant No. 2021B0101220004.

References

1. Abdel-Basset, M., Saleh, M., Gamal, A., Smarandache, F.: An approach of TOPSIS technique for developing supplier selection with group decision making under type-2 neutrosophic number. Appl. Soft Comput. **77**, 438–452 (2019)

2. Beni, S.A., Sheikh-El-Eslami, M.K.: Market power assessment in electricity markets based on social network analysis. Comput. Electr. Eng. **94**, 107302 (2021)
3. Biswas, K., Biswas, S., Sen, P.: Block size dependence of coarse graining in discrete opinion dynamics model: Application to the us presidential elections. Physica A: Statistical Mechanics and its Applications **566**, 125639 (2021)
4. Blondel, V.D., Hendrickx, J.M., Tsitsiklis, J.N.: On krause's multi-agent consensus model with state-dependent connectivity. IEEE Trans. Autom. Control. **54**(11), 2586–2597 (2009)
5. Cabrerizo, F.J., Al-Hmouz, R., Morfeq, A., Balamash, A.S., Martínez, M.Á., Herrera-Viedma, E.: Soft consensus measures in group decision making using unbalanced fuzzy linguistic information. Soft Comput. **21**(11), 3037–3050 (2017)
6. Chao, X., Kou, G., Peng, Y., Herrera-Viedma, E., Herrera, F.: An efficient consensus reaching framework for large-scale social network group decision making and its application in urban resettlement. Information Sciences **575**, 499–527 (2021)
7. Cheng, C., Yu, C.: Opinion dynamics with bounded confidence and group pressure. Physica A: Statistical Mechanics and its Applications **532**, 121900 (2019)
8. (Director), C.N.: Inception. United States: Warner Bros. (2010)
9. Dong, Y., Zha, Q., Zhang, H., Kou, G., Fujita, H., Chiclana, F., Herrera-Viedma, E.: Consensus reaching in social network group decision making: Research paradigms and challenges. Knowl. Based Syst. **162**, 3–13 (2018)
10. Dong, Y., Zhan, M., Kou, G., Ding, Z., Liang, H.: A survey on the fusion process in opinion dynamics. Inf. Fusion **43**, 57–65 (2018)
11. Douven, I., Hegselmann, R.: Mis- and disinformation in a bounded confidence model. Artif. Intell. **291**, 103415 (2021)
12. Hashemi, E., Pirani, M., Khajepour, A., Fidan, B., Kasaiezadeh, A., Chen, S.: Opinion dynamics-based vehicle velocity estimation and diagnosis. IEEE Trans. Intell. Transp. Syst. **19**(7), 2142–2148 (2018)
13. Hassani, H., Razavi-Far, R., Saif, M., Chiclana, F., Krejcar, O., Herrera-Viedma, E.: Classical dynamic consensus and opinion dynamics models: A survey of recent trends and methodologies. Inf. Fusion **88**, 22–40 (2022)
14. Hua, Z., Jing, X., Martínez, L.: Consensus reaching for social network group decision making with ELICIT information: A perspective from the complex network. Inf. Sci. **627**, 71–96 (2023)
15. Huang, D.W., Yu, Z.G.: Dynamic-sensitive centrality of nodes in temporal networks. Scientific reports **7**(1), 1–11 (2017)
16. Kawasaki, T., Wada, R., Todo, T., Yokoo, M.: Mechanism design for housing markets over social networks. In: Dignum, F., Lomuscio, A., Endriss, U., Nowé, A. (eds.) AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021. pp. 692–700. ACM (2021)
17. Lu, Y., Xu, Y., Herrera-Viedma, E., Han, Y.: Consensus of large-scale group decision making in social network: the minimum cost model based on robust optimization. Inf. Sci. **547**, 910–930 (2021)
18. Ma, Q., Qin, J., Anderson, B.D., Wang, L.: Exponential consensus of multiple agents over dynamic network topology: Controllability, connectivity, and compactness. IEEE Transactions on Automatic Control pp. 1–16 (2023). https://doi.org/10.1109/TAC.2023.3245021
19. Nedić, A., Olshevsky, A., Rabbat, M.G.: Network topology and communication-computation tradeoffs in decentralized optimization. Proceedings of the IEEE **106**(5), 953–976 (2018)

20. Silver, D., Singh, S., Precup, D., Sutton, R.S.: Reward is enough. Artif. Intell. **299**, 103535 (2021)
21. Sun, X., Qiu, J.: Two-stage volt/var control in active distribution networks with multi-agent deep reinforcement learning method. IEEE Trans. Smart Grid **12**(4), 2903–2912 (2021)
22. Ureña, R., Chiclana, F., Melançon, G., Herrera-Viedma, E.: A social network based approach for consensus achievement in multiperson decision making. Inf. Fusion **47**, 72–87 (2019)
23. Ureña, R., Kou, G., Dong, Y., Chiclana, F., Herrera-Viedma, E.: A review on trust propagation and opinion dynamics in social networks and group decision making frameworks. Inf. Sci. **478**, 461–475 (2019)
24. Wang, H., Shang, L.: Opinion dynamics in networks with common-neighbors-based connections. Physica A: Statistical Mechanics and its Applications **421**, 180–186 (2015)
25. Weng, T., Dvijotham, K.D., Uesato, J., Xiao, K., Gowal, S., Stanforth, R., Kohli, P.: Toward evaluating robustness of deep reinforcement learning with continuous control. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020)
26. Xie, G., Chen, J., Li, Y.: Hybrid-order network consensus for distributed multi-agent systems. Journal of Artificial Intelligence Research **70**, 389–407 (2021)
27. Xie, G., Xu, H., Li, Y., Hu, X., Wang, C.D.: Fast distributed consensus seeking in large-scale and high-density multi-agent systems with connectivity maintenance. Information Sciences **608**, 1010–1028 (2022)
28. Xie, G., Xu, H., Li, Y., Wang, C.D., Zhong, B., Hu, X.: Consensus seeking in large-scale multiagent systems with hierarchical switching-backbone topology. IEEE Transactions on Neural Networks and Learning Systems pp. 1–15 (2023). https://doi.org/10.1109/TNNLS.2023.3290015
29. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: Lstm cells and network architectures. Neural computation **31**(7), 1235–1270 (2019)
30. Zhang, H., Dong, Y., Herrera-Viedma, E.: Consensus building for the heterogeneous large-scale GDM with the individual concerns and satisfactions. IEEE Trans. Fuzzy Syst. **26**(2), 884–898 (2018)
31. Zhang, H., Dong, Y., Xiao, J., Chiclana, F., Herrera-Viedma, E.: Consensus and opinion evolution-based failure mode and effect analysis approach for reliability management in social network and uncertainty contexts. Reliab. Eng. Syst. Saf. **208**, 107425 (2021)
32. Zhang, K., Yang, Z., Başar, T.: Multi-agent reinforcement learning: A selective overview of theories and algorithms. Handbook of Reinforcement Learning and Control pp. 321–384 (2021)
33. Zhang, T., Ye, Q., Bian, J., Xie, G., Liu, T.: MFVFD: A multi-agent q-learning approach to cooperative and non-cooperative tasks. In: Zhou, Z. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021. pp. 500–506. ijcai.org (2021)
34. Zhu, L., He, Y., Zhou, D.: Neural opinion dynamics model for the prediction of user-level stance dynamics. Inf. Process. Manag. **57**(2), 102031 (2020)