Citation information: DOI: 10.1109/TNNLS.2025.3588236, IEEE Transactions on Neural Networks and Learning Systems

Multiple Influences Maximization Under Dynamic Link Strength in Multi-Agent Systems: The Competitive and Cooperative Cases

Mincan Li, Zidong Wang, *Fellow, IEEE*, Simon J. E. Taylor, Kenli Li, Xiangke Liao, and Xiaohui Liu

***Abstract*—This paper addresses the issue of multiple influences maximization under the condition of dynamic link strength (MIMDLS) in multi-agent systems (MASs). Initially, a novel model for dynamic link strength within MASs is suggested to facilitate the simulation of multiple influences diffusion. Subsequently, the MIMDLS problem is formulated with both the competitive and cooperative scenarios being examined. In response, two diffusion models, specifically the competitive multiple influences independent cascade (Cp-MIIC) model and the cooperative multiple influences linear threshold (Cr-MILT) model, are designed for MASs. Furthermore, a distributed deep reinforcement learning (DRL) framework is established based on MASs by incorporating asynchronous training and updating processes for seed selection in the context of multiple influences. Moreover, the developed distributed DRL algorithm encompasses the estimation of Q-value as well as the management of constraints within the Cp-MIIC and Cr-MILT models. Finally, comprehensive experiments are conducted to 1) validate the effectiveness and efficiency of the proposed models and algorithms in terms of multiple influence diffusion; and 2) benchmark their performance against state-of-the-art methods.**

***Index Terms*—Multiple information maximization, influence diffusion, multi-agent systems, dynamic link strength, deep reinforcement learning.**

I. Introduction

THE problem of influence maximization (IM), initially introduced in [16], has emerged as a focal point in research on viral marketing, link prediction [18], [60], and community detection within social networks (SNs). At its core, IM seeks to harness the potential of influencers within a static network to propagate the influence of a specific topic or product, thereby expanding its acceptance among an increasing number of ordinary users. The objective of IM is to find the seed users who can maximize the number of ordinary users influenced, typically under a specific diffusion model. The IM problem can be considered an algorithmic problem or tackled as a discrete optimization problem, and it has been proven to be NP-hard under traditional diffusion models [25].

IM-related research has predominantly focused on three principal strategies: approximation algorithms, heuristic methods, and community-based approaches. Approximation algorithms, which tackle IM as a combinatorial optimization [52], [55], [59], [63] challenge, have offered provable guarantees, demonstrating a $(1-1/e)$ ratio based on the greedy principle within the Independent Cascade (IC) model and Linear Threshold (LT) model. Consistent with these findings, an extensive and expanding corpus of studies has explored optimal greedy techniques to achieve the best possible solutions. Heuristic methods, on the other hand, have been favored for their scalability and faster execution time, as they do not necessitate the computation of approximation bounds. Notably, metaheuristic algorithms have simplified the complexity to $O(kd(m+n))$ in the context of the IC model [21], where k is the size of seed set, d is the length of deep searching, m is the total number of edges and n is the population of network. The community-based approach has been introduced as a means of discovering superior solutions in comparison to some advanced heuristic methods [10]. Moreover, integrated strategies of greedy heuristic and Hop-based approaches have been proposed, yielding satisfactory outcomes across various scenarios of information diffusion [47].

Over time, to better align with real-world scenarios, a diverse array of IM variants has been introduced in the literature [5], [37], [39], and some of them are within dynamic environments [1], [9]. It is important to note that, most of studies [9], [35], [39] have concentrated on the availability of links, they have largely overlooked the dynamics associated with link weights. In [39], a novel sketch-based method has been introduced by employing an index that adjusts sketches through expanding or shrinking them, thereby facilitating the determination of k-coverage within a dynamic network [39]. Additionally, the seed set and pseudo-seed set have been designed to support T×oneHop approach to deal with changing links among SNs [35].

Recently, significant emphasis has been placed on developing behavior-aware IM strategies in SNs, recognizing that effective IM solutions should account for more than just the graph structures. In fact, it is crucial to incorporate user behaviors, activities, and their variations into the analysis. Behavior-

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB2104000, the National Natural Science Foundation of China under Grants 61625202, 61751204, 61860206011 and 62206091, the Natural Science Foundation of Hunan Province of China under Grant 2023JJ40166, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany. *(Corresponding author: Kenli Li.)*

Mincan Li and Kenli Li are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China; and also with the National Supercomputing Center in Changsha, Changsha, Hunan 410082, China. (emails: limc@hnu.edu.cn; lkl@hnu.edu.cn)

Zidong Wang, Simon J. E. Taylor and Xiaohui Liu are with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom. (emails: Zidong.Wang@brunel.ac.uk; Simon.Taylor@brunel.ac.uk; Xiaohui.Liu@brunel.ac.uk)

Xiangke Liao is with the Collaborative Innovation Center of High Performance Computing, National University of Defense Technology, Changsha 410073, China. (email: xkliao@nudt.edu.cn).

aware IM examples, such as the label-aware model [4], target-aware model [29], and topic-aware model [49], underscore the pivotal influence of user behaviors on the spread of influence. By extracting and examining user behavior characteristics, the IM problem can be approached more effectively. A practical method involves the creation of a multi-agent system (MAS) model, which simulates user activities and manages behavioral attributes to enhance the process of influence diffusion.

Extensive research has been conducted on IM and its various applications, then the concept of multiple influences maximization (MIM) [45] has been explored by a growing number of researchers. An example in Fig.1 is illustrated for a common MIM problem in a directed graph. Three influences are ready for diffusion and marked as three colors. Three diffusion probabilities are set on every edge for the corresponding influence and marked as $p\omega_1, p\omega_2, p\omega_3$. Typically, the constraint on the number of times a node can be activated is not required in a general MIM. Thus, in a general MIM problem, a node can be successfully activated by more than one influence. With the constraints that the intersection of the seed sets of different influences should be an empty set and the total seed selection should be limited in *k*, the objective of a general MIM is to select *k* seed nodes to maximize the final diffusion spread. In Fig. 1, the optimal seed set had been selected (*k*=3): three nodes are selected as seed nodes for the first influence (blue), the second influence (orange), and the third influence (green) respectively. By diffusing 3 influences from 3 seed nodes with corresponding probabilities, the final influence spread is 14 (5 orange, 5 green, and 4 blue nodes).

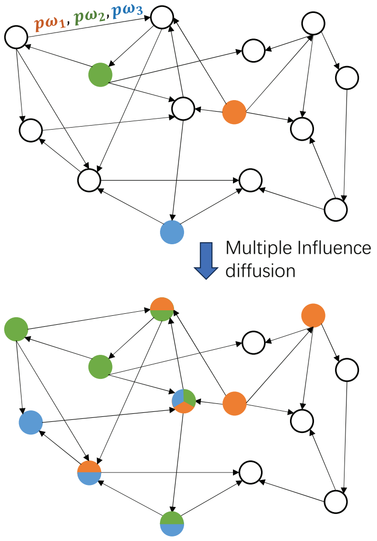

$p\omega_1, p\omega_2, p\omega_3$
Multiple Influence diffusion

Fig. 1. An example of the general MIM problem (k=3).

Machine learning technologies have strong advantages of efficiency and generalization ability to solve the MIM problem, especially Reinforcement Learning (RL) methods [33]. MIM can be solved by formulating multiple decision-making sequences in discrete space using RL approaches. Furthermore, the solution of MIM can be extended to the large-scale and complicated networks via Deep RL because of its characteristic offline training and online decision-making. Thus, the RL-based method has an absolute advantage in handling the complexity of dynamic factors in link properties.

The literature on MIM had been focused on fixed topologies [53], multi-round influence diffusion [34] and multiple networks [44], while ignoring dynamic factors such as changing link weights. This oversight can largely be attributed to the fact that traditional IM and MIM investigations have tended to prioritize network topologies over the nuances of link properties and user effects on links. For instance, to maximize the total influence in multiple isomorphic networks, parallel greedy framework had been provided [53]. Considering competitive commodities in real-world, a model for competitive influence diffusion had been suggested: strategies involving known and unknown competitors have been utilized, leading to the formulation of an *n*-player diffusion game that aims for a Nash equilibrium [32], [62]. Nonetheless, MIM is distinctively challenged by the prevalent competitive and cooperative/coordinate relationships among influences under a dynamic circumstance, presenting a complex problem that has begun to draw significant scholarly attention.

This paper explores the problem of multiple influences maximization under the condition of dynamic link strength (MIMDLS) in MASs with applications to SNs. The key contributions are outlined as follows.

1) A novel dynamic model is proposed for link strength, grounded in MAS, encompassing the generation of link strength, interaction protocols among agents, and the evolving rules for dynamic link strength.
2) The concept of the MIMDLS problem is formulated to model and encapsulate the process of maximizing multiple influences within a social network, which is further diversified into the cases of Competitive MIMDLS (Cp-MIMDLS) and Cooperative MIMDLS (Cr-MIMDLS).
3) Based on MASs, a distributed deep reinforcement learning (DRL) approach is developed for MIM with aim to estimate and ensure the optimization of the seed set, while addressing the complexities associated with MIMDLS scenarios.

The structure of this paper is organized in the following manner. Section II delves into the current state-of-the-art technologies pertinent to the IM problem within the context of SNs. Section III lays out the basic definitions related to the MIM problem and its derivative cases as they apply to SNs, along with relevant preliminary concepts. In Section IV, the communication protocols, a dynamic model for link strength within MAS, and the diffusion models are thoroughly presented. Section V is dedicated to showcasing a variety of experimental outcomes, analyses of parameters, and the juxtaposition of the proposed methodologies against a range of standard benchmarks. The paper is concluded in Section VI, where summative remarks are made and prospective avenues for future investigation are identified.

II. Related Work

A multitude of past studies has established that the static IM problem is an NP-hard problem under various diffusion models, including the IC model, LT model, Triggering (TR)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TNNLS.2025.3588236, IEEE Transactions on Neural Networks and Learning Systems

model, and Continuous-Time (CT) model. Within these investigations, a greedy algorithm has often been employed to select seed nodes up to a specified k parameter limit. This greedy approach is underpinned by the theoretical assurance provided by a non-negative monotone sub-modular function. The methodologies that utilize this greedy framework can be broadly categorized into three groups: simulation-based [61], proxy-based [20], and sketch-based [11] approaches (the estimations of influence diffusion is computed by generating several sketches under a specific diffusion model). Despite their theoretical underpinnings, these greedy methodologies have been criticized for their extensive computational time and diminished efficiency in influence spread, particularly within large network structures. The trade-off between influence diffusion and computational efficiency has been somewhat mitigated by the adoption of meta-heuristic algorithms [51]. For instance, a genetic algorithm has been introduced to refine seed selection through various strategies of population initialization [13]. Nevertheless, the heuristic approaches lack a theoretical foundation and necessitate algorithmic design tailored to the specific diffusion model in use.

Contrary to the static IM, dynamic influence maximization (DIM) faces the significant challenge of constantly evolving user topologies. Recent studies have predominantly focused on evolutionary computation methods to address the complexities inherent in DIM. One such approach is an adaptive evolutionary method that enhances the candidate solution by pinpointing users with significant influencing capabilities [31]. This method emphasizes the identification of potential influencers as a core component of the algorithm, serving as an alternative to the sketch-based approach in accommodating dynamic network changes. Additionally, from the standpoint of link structures, the notion of an “effective link” has been introduced to lay the groundwork for a two-stage IM algorithm, which delves into the exchange of information between user pairs, aiming to refine the selection process for seed users and thereby enhance the overall quality of the influence network [19]. On the other hand, inspired by clustering concepts, the original network has been transformed into one of coarser granularity, and the DIM problem has been approached by identifying seed users through the lens of community structure information [41]. Despite this innovative approach, the dynamic weight of links, a crucial determinant of influence diffusion, has often been overlooked. The current methodologies that focus on dynamic links fall short in networks characterized by dynamic link strength/weight, which is due mainly to the fact that the dynamics in such networks are deeply influenced by user behaviors, interactions, and preferences, which cannot be adequately addressed using standard evolutionary methods or clustering strategies.

In the realm of prior research, some efforts have been made for addressing the MIM problem within competitive settings with aim to resolve real-world challenges. One such approach is a maximization algorithm designed to circumvent competitive nodes by taking into account community dispersion and dynamic attributes aligned with user interests [50]. An IM method that focuses on the examination of homogeneous communities and the impact of inactive nodes has been deployed in [54] to assess weak influence among potential nodes within SNs. Furthermore, a competitive version of the LT model has been developed for MIM, which assigns a dimensional vector to each user to track the influence probability of various types [6]. In an effort to consider the intricate interplay among influences (including both competitive and complementary dynamics), a deep recursive hybrid model has been introduced for assessing the probabilities of influence between node pairs concerning products [23]. While these models offer potential solutions for the MIM challenge, there has been scant exploration into MIM within the context of dynamic link strength, which is an area that warrants urgent and thorough investigation.

The utility and impact of learning-based methodologies extend across a wide array of IM challenges, with RL frameworks [12], [46] standing out as particularly effective and efficient solutions [17], [33], [48]. RL approaches [27], [57] are especially well-suited to IM issues especially when these are conceptualized as combinatorial optimization problems. The adaptation of IM into a Markov Decision Process (MDP) framework [22], [24], [56], followed by the application of RL [15], [58] to assimilate behaviors from historical network topologies, has been employed to tackle the contingency-aware IM problem [7]. In a distinct approach, an orthogonal paradigm has been developed to predict expected influence diffusion using an RL algorithm, which notably obviates the need for building the model from the ground up [28]. Furthermore, Graph Neural Networks (GNNs) have been brought into the fold to aid in addressing IM and its related applications. A position-aware inductive GNN model has been designed to focus on the encoding of local neighborhood structures, which leverages a set of anchor nodes to capture the positions of all nodes within the network, thereby optimizing global reachability [38]. A wide array of models specifically designed for various IM variants have been developed to suit particular situations via graph computation [8], [43]. The Celfie method, for example, has been explored to bypass the constraints inherent in conventional diffusion models by extracting influence representations through the analysis of diffusion cascade information [40]. An adversarial graph embedding technique has been implemented to address the fairness in nodes’ influenceability by sensitive attributes, which involves the creation of a discriminator for sensitive attribute recognition while simultaneously training a graph embedding auto-encoder [26]. Therefore, integrating RL and GNN advantages constitutes an effective and practical solution for addressing the complexities of dynamic link strength within MIM challenges.

III. Problem Formulation

In this section, the foundational concepts related to IM are introduced, and the MIMDLS problem is defined along with its associated two cases.

A. Preliminaries

Multiple Influences Maximization (MIM) problem: Given an SN with m $(m > 1)$ types of influences, the objective is to identify k seed nodes that maximize the total number of activated nodes influenced by all types of influences. The SN is denoted as $G = \{V, E, P\}$, where V and E represent the sets of nodes and edges, respectively, and P is the set of probabilities indicating the diffusion likelihood of influences between nodes. For each influence type i ($1 \leq i \leq m$) and any two nodes u and v in G, $p_i(u, v) \in P$ signifies the diffusion probability of the i^{th} influence from node u to node v. The goal is to construct a seed set $S = S_1 \cup S_2 \cup \cdots \cup S_m$ ($|S| = k$), ensuring that $S_1 \cap S_2 \cap \cdots \cap S_m = \varnothing$, with each S_i being the seed set for the i^{th} influence, and the total number of seeds across all sets equal to k.

Basic IC model: The basic IC model considers the diffusion of a single type of influence within a given SN $G = \{V, E, P\}$. Initially, all nodes within the seed set are activated at the first time step. Subsequently, at the second time step, each activated node has the potential to influence its neighboring nodes based on the diffusion probability associated with the respective link. The process permits only the nodes activated in the current time step to influence others in the subsequent step. This diffusion continues until no additional nodes can be activated, signifying the end of the process.

Basic LT model: The basic LT model addresses the spread of a single type of influence among users in an SN $G = \{V, E, P\}$. Each node within this network is assigned a threshold $\theta \in (0, 1)$. The diffusion process initiates from seed nodes in an activated state, diffusing the influence to neighboring nodes according to the specified probabilities of the corresponding edges. A node becomes activated when the cumulative influence it receives reaches or exceeds its threshold θ. The diffusion process concludes when there are no further nodes that can be activated.

B. Problem statement

Link Strength (LS): In this paper, link strength is synonymous with diffusion probability.

1) For a directed graph, the link strength of the edge from node u to node v, denoted as (u, v), is updated by user u and represented as $ls(u, v)$, where $ls(u, v) = \{ls_i(u, v)\}$ ($1 \leq i \leq m$). Here, $ls_i(u, v)$ signifies the strength of the i^{th} influence as it diffuses from node u to node v, with its value ranging within $[0, 1]$.
2) In the case of an undirected graph, the link (u, v) is treated as two directed links: (u, v) and (v, u). Accordingly, the link strengths for the link (u, v) are given by $ls(u, v)$, which is updated by user u, and $ls(v, u)$, which is under the control of user v, as illustrated in Fig. 2.

It should be noted that $ls_i(u, v) \in [0, 1]$, representing the strength of the i^{th} influence from node u to node v, can be understood in three distinct scenarios at the current time step:

1) If the link strength is 0, it indicates the absence of a link from node u to node v.
2) In the event that user u became an activated node in the last time step, node v will be successfully activated when $ls_i(u, v) = 1$.
3) Should user u have been activated in the last time step, the i^{th} influence will be diffused to user v with a probability equal to $ls_i(u, v)$ that is in (0,1).

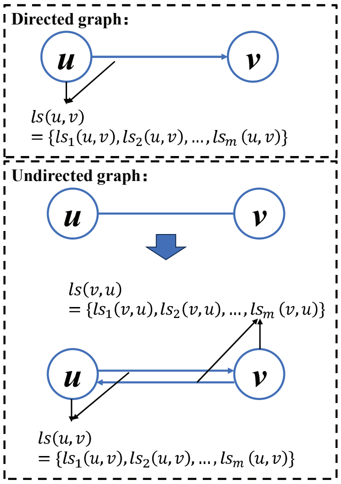

Directed graph:
u
v
$ls(u,v)$
$= \{ls_1(u,v), ls_2(u,v), \ldots, ls_m(u,v)\}$
Undirected graph:
u
v
$ls(v,u)$
$= \{ls_1(v,u), ls_2(v,u), \ldots, ls_m(v,u)\}$
u
v
$ls(u,v)$
$= \{ls_1(u,v), ls_2(u,v), \ldots, ls_m(u,v)\}$

Fig. 2. An illustration for link strength.

Moreover, a link strength of 0 signifies the dissolution of the relationship between users, whereas a link strength of 1 represents a scenario of "blind following action".

MIM under Dynamic Link Strength (MIMDLS): In the context of an SN where m ($m > 1$) types of influences are present, the challenge is to identify k seed nodes that will maximize the total number of activated nodes under the condition that the link strength set LS is dynamic and subject to updates by users at each time step. The SN with LS is represented as $G' = \{V, E, LS\}$, with $LS = \{ls(u, v)\}$ ($u, v \in V$) encompassing all link strengths in the network, where $ls(u, v) = \{ls_i(u, v)\}$ ($1 \leq i \leq m$) for each link from node u to node v. These link strengths $ls(u, v)$ are dynamically updated by the initiating user u throughout the diffusion process. The composition of the seed set must satisfy $S = S_1 \cup S_2 \cup \cdots \cup S_m$ ($|S| = k$), ensuring that $S_1 \cap S_2 \cap \cdots \cap S_m = \varnothing$, with $S_1, S_2, ..., S_m$ designated as the seed sets corresponding to the $1^{st}, 2^{nd}, ..., m^{th}$ influences, respectively.

The dynamic nature of link strength updates in the MIMDLS framework is influenced by user communications and interactions, reflecting the variable acceptability of influences by users. It should be mentioned that every user is a potential node for every influence, and when a user's link strength is 0, it does not mean this user is a repelling user; it only indicates that the user's acceptability is 0 at the current time step and the acceptability could be updated by the user in the future. Within the MIMDLS context, the key elements driving diffusion maximization include the dynamic modification of link weights and the interplay among the m types of influences. In light of the complex dynamics among multiple influences, two distinct cases of MIMDLS are identified, namely, competitive and cooperative cases, which are detailed as follows.

Competitive MIMDLS (Cp-MIMDLS): In Cp-MIMDLS, the m types of influences present in the MIMDLS scenario are considered to be in competition with each other, meaning a node once activated by one influence cannot be influenced by another. The challenge in Cp-MIMDLS is to identify a seed set $S = S_1 \cup S_2 \cup \cdots \cup S_m$ ($|S| = k$), where each node is activated by only one type of influence, aiming to maximize the total number of nodes activated across all influences.

Cooperative MIMDLS (Cr-MIMDLS): In Cr-MIMDLS, the m types of influence within MIMDLS interact cooperatively, allowing a node to be activated by multiple influences, contrasting with Cp-MIMDLS's competitive nature. In this cooperative setting, successful activation by one influence enhances the activation probability for others. Cr-MIMDLS aims to maximize the total successful activations under two conditions: 1) the seed set $S = S_1 \cup S_2 \cup \cdots \cup S_m$ $(|S| = k)$ is composed of subsets for each influence, and 2) each subset S_i contains an equal fraction of the total seeds, with $|S_1| = |S_2| = \cdots = |S_m| = k/m(k\%m = 0)$. The reason for setting the subsets to the same size is to avoid the impact of initial diffusion advantage on the MIM result and to ensure diffusion fairness among influences.

User/Agent: In an MAS model for MIMDLS, each user is modeled as an agent $a_j = \{LS^j, F^j, ST^j\}$, with j serving as the agent's identifier. Every agent a_j possesses a link strength matrix $LS^j = \{ls_i(a_j, a_q)\}$ $(1 \leq i \leq m, q \in N_{a_j})$, where N_{a_j} represents the set of neighbor identifiers for a_j. The feature matrix for m influences is denoted by $F^j = \{\vec{f}_1^j, \vec{f}_2^j, \cdots, \vec{f}_m^j\}$, where $\vec{f}_1^j$ is a row vector that captures the behavioral features of the 1^{st} influence on a_j, and the dimensionality of $\vec{f}_1^j$, represented by d_f, is determined by the database. The state set $ST^j = \{st_1^j, st_2^j, \cdots, st_m^j\}$ $(st_i^j = 0 \text{ or } 1, |ST^j| = m)$ indicates the activation status of a_j with respect to each influence; $st_i^j = 1$ signifies that a_j is activated by the i^{th} influence, while $st_i^j = 0$ indicates unsuccessful activation.

The initial state set for agent a_j is set as $ST^j = \{0, \cdots, 0\}(|ST^j| = m)$ before influence diffusion. It should be mentioned that in a competitive scenario, the value of $\sum_{i=1}^{m} st_i^j$ (of the state set ST^j) is either 1 or 0, indicating that an agent can be activated by at most one type of influence or not activated at all. Conversely, in a cooperative scenario, $\sum i = 1^m st_i^j$ can be 0, indicating no activation, or $\sum i = 1^m st_i^j \geq 1$, allowing for the possibility of an agent being activated by one or more types of influence.

IV. Main Model and Approach

This section introduces diffusion models for multiple influences in both competitive and cooperative environments and establishes the principal MAS model for dynamic link strength, including interaction rules for MAS diffusion. It concludes with the design of a distributed RL framework tailored for the MIM solution. The relationships among the proposed problems and main models are depicted in Fig. 3: The multiple influences diffusions of competitive and cooperative MIMDLS problems are simulated through MAS based on diffusion models (Cp-MIIC and Cr-MILT), and the dynamic link strength is implemented trough agent communications in MAS simultaneously. Eventually, MIMDLS is solved by a distributed RL framework based on MAS through interactions with the simulation environment.

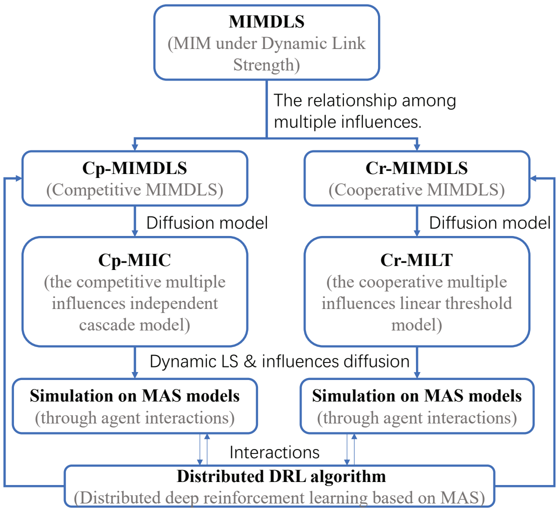

MIMDLS
(MIM under Dynamic Link Strength)
The relationship among multiple influences.
Cp-MIMDLS
(Competitive MIMDLS)
Cr-MIMDLS
(Cooperative MIMDLS)
Diffusion model
Diffusion model
Cp-MIIC
(the competitive multiple influences independent cascade model)
Cr-MILT
(the cooperative multiple influences linear threshold model)
Solve
Solve
Dynamic LS & influences diffusion
Simulation on MAS models
(through agent interactions)
Simulation on MAS models
(through agent interactions)
Interactions
Distributed DRL algorithm
(Distributed deep reinforcement learning based on MAS)

Fig. 3. The relationships among proposed problems and models.

A. Multiple influences diffusion models based on MAS

Building upon the conventional diffusion models IC and LT, two novel diffusion models are tailored to accommodate dynamic link strength and multiple influences within both competitive and cooperative settings: 1) the competitive multiple influences independent cascade (Cp-MIIC) model, and 2) the cooperative multiple influences linear threshold (Cr-MILT) model. It should be noted that, the design of the influence diffusion rules determines the environment to which the model can adapt. In other words, a competitive/cooperative model can be designed based on the LT/IC models. The reason for designing the Cp-MIIC and Cr-MILT models is that it is much easier to understand the real-world physical meaning (of cooperation and competition) reflected by these two models.

1) Cp-MIIC model: Upon the initial activation of nodes in the seed set S with m types of influence, these activated nodes proceed to activate their neighboring nodes according to probabilities matching the respective link strengths. Activation in subsequent time steps is limited to nodes that were activated in the immediate preceding step. The Cp-MIIC model diverges from the standard IC model by adhering to the following specific rules:

i. Once activated, nodes are immune to further external influences.
ii. An activated node can propagate only the type of influence that led to its activation.
iii. While inactive nodes may receive multiple influences from various neighbors, they can ultimately be activated by only a single type of influence.

The distinctive aspect of the Cp-MIIC model is encapsulated in rule iii, which mandates that if an inactive node is subjected to multiple influences, these influencing neighbors are arranged in a sequence based on the strength of their links. The influencing process then follows this sequence, with each neighbor attempting to activate the inactive node using its influence until the node is either activated or all potential influencers have been considered. Overall, the same constraint is reflected by the three rules: a node can be successfully

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TNNLS.2025.3588236, IEEE Transactions on Neural Networks and Learning Systems

activated by only one influence, indicating a strict competitive relationship among influences.

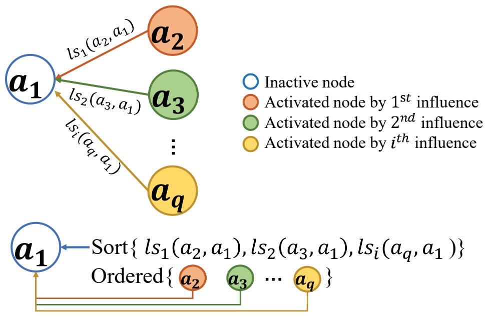

$ls_1(a_2, a_1)$
$ls_2(a_3, a_1)$
$ls_i(a_q, a_1)$
a_1
a_2
a_3
a_q
Inactive node
Activated node by 1^{st} influence
Activated node by 2^{nd} influence
Activated node by i^{th} influence
Sort{ $ls_1(a_2, a_1), ls_2(a_3, a_1), ls_i(a_q, a_1)$ }
Ordered{ a_2 a_3 ··· a_q }

Fig. 4. An example of Cp-MIIC model.

To demonstrate the Cp-MIIC model, consider a_1 in Fig. 4 as an example of an inactive node receiving multiple types of influence at a single time step. The $1^{st}, 2^{nd}, \cdots, i^{th}$ influences are simultaneously disseminated by a_1's activated neighbors $a_2, a_3, \cdots, a_q$, all of which are adjacent to a_1. These neighbors are ranked according to their link strengths with a_1, arranged in descending order as $a_2, a_3, \cdots, a_q$, as depicted at the bottom of Fig. 4. Following this sequence, the neighbors attempt to activate a_1 with the probabilities corresponding to their link strengths. If a_1 is successfully activated by the i'^{th} influence from one of its neighbors, it will set $st^1_{i'}$ (where $st^1_{i'} \in ST^1$) to 1, indicating activation, and will then proceed to influence its own inactive neighbors in the next time step. If, however, none of a_1's neighbors manage to activate it, a_1 will remain inactivate state($|ST^1| = 0$) until the beginning of the next time step, and it may be activated again by other agents in the next step.

2) Cr-MILT model: In the Cr-MILT model, similar to the conventional LT model, each agent a_j is assigned a unique threshold vector θ^j, expressed as $\theta^j = \{\theta^j_0, \theta^j_1, \cdots, \theta^j_m\}$. The values $\theta^j_1, \cdots, \theta^j_m$ fall within the interval $(0, 1)$ and represent the activation thresholds for each corresponding type of influence impacting a_j. Distinctively, $\theta^j_0 \in [1, m]$ specifies the type threshold for influences on a_j, indicating that a_j can be activated by up to θ^j_0 different types of influence.

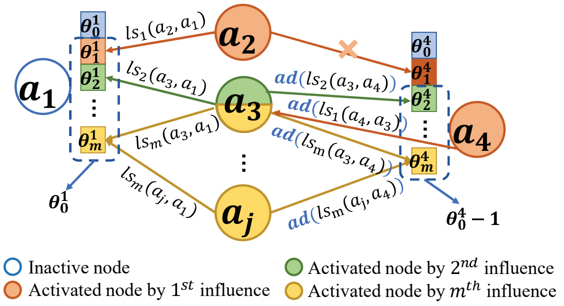


Fig. 5. An example of Cr-MILT model.

Unlike the traditional LT model, the Cr-MILT model incorporates a cooperation vector for influences, denoted as $cv = \{c_1, c_2, \cdots, c_m\}$, where each c_i signifies the enhancement index for the i^{th} influence. For instance, if agent a_j is already activated by the 5^{th} influence, the link strengths for all other types of influence (excluding the 5^{th}) will receive an enhancement indexed by c_5 at the subsequent time step. (The method for calculating the enhancement index c_i will be elaborated later). The diffusion mechanism in Cr-MILT is characterized by two situations: 1) the activation of an inactive node, and 2) the re-activation of an already activated node by additional influences. These processes are visually represented in Fig. 5.

Situation 1: Here, a_1 is initially inactive, with its type threshold of influences denoted as θ^1_0. At each time step, a_1 aggregates the influence values from each type of influence exerted by its neighbors, comparing the aggregate with its respective activation thresholds. For instance, the total influence for the m^{th} type on a_1 at a given time step might be calculated as $ac\theta^1_m = ls_m(a_3, a_1) + ls_m(a_j, a_1)$. The comparison between θ^1_m and $ac\theta^1_m$ is then conducted according to

$$eva(\theta^1_m, ac\theta^1_m) = (ac\theta^1_m - \theta^1_m)/\theta^1_m \tag{1}$$

The extent to which the accumulated threshold exceeds original threshold is reflected by $eva(.,.)$. The node cannot be activated successfully when $eva(.,.) < 0$. The evaluation is repeated for each type of influence, resulting in a set of evaluations $eva(\theta^1_i, ac\theta^1_i)$ for $1 \leq i \leq m$, which are then arranged in descending order. The top θ^1_0 values from this ordered list are compiled into an EVA queue for a_1. Subsequently, a_1 is activated by the types of influence corresponding to all positive entries in the EVA queue.

Situation 2: In this situation, a_4 is already activated by the 1^{st} influence, meaning it can no longer be influenced by the 1^{st} influence from a_2, but it remains susceptible to activation by up to $\theta^4_0 - 1$ other types of influence. At the current time step, a_4 receives the 2^{nd} and m^{th} influences through three connections, each with its own link strength. Given the cooperative nature of the influences and the enhancement index c_1 for the 1^{st} influence, the link strengths ($ls_2(a_3, a_4)$, $ls_m(a_3, a_4)$, and $ls_m(a_j, a_4)$) are adjusted as per the equation below:

$$ad(ls_i(a_j, a_{j'})) = ls_i(a_j, a_{j'})(1 + \frac{1}{n}\sum c_p) \tag{2}$$

Here, c_p denotes the enhancement index for the p^{th} influence, with p belonging to the set $NA_{a_{j'}}$ that comprises the identifiers of influences that have successfully activated $a_{j'}$. The set $NA_{a_{j'}}$, with cardinality n, represents the influences that have activated $a_{j'}$. For example, the adjusted link strength from a_3 to a_4 for the 2^{nd} influence is calculated as $ad(ls_2(a_3, a_4)) = ls_2(a_3, a_4)(1 + c_1)$. Similarly, the adjustment from a_4 to a_3 for the 1^{st} influence is determined by $ad(ls_1(a_4, a_3)) = ls_1(a_4, a_3) * (1 + 1/2(c_2 + c_m))$, employing an average of the enhancement indices for the calculation. After these link strength adjustments, the diffusion process proceeds analogously to Situation 1). It should be note that Eq.(2) is used for adjusting diffusion probability at the current time step under Cr-MILT model but not real change the value of LS of the nodes. In other words, a temporary value is calculated by Eq.(2) and is used for threshold accumulation during the diffusion process under the Cr-MILT model.

B. MAS model for dynamic link strength

In the MAS model for MIM, agents representing users are interconnected within an SN denoted as G. Dynamically updating link strength will be executed by agents before influence diffusion at each time step, and prior to that, the initialization of link strengths should be designed. The initialization of link strengths is a critical step to facilitate dynamic link strength implementation, acknowledging that each user exhibits distinct behavioral features under each type of influence, which in turn inform the initial link strengths derived from feature vectors. The initialization depends on agent communications and observations before the process of influence diffusion. The process starts by calculating the attention index α^i_{jq} of every user a_j towards each neighbor a_q ($q \in N_{a_j}$) under the i^{th} influence, as shown in Eq. (3):

$$\alpha^i_{jq} = [1, 1, \cdots, 1]\vec{f}_i^{j\top} \parallel \vec{f}_i^{q\top} \tag{3}$$

Here, α^i_{jq} represents the attention index of user a_j towards its neighbor a_q under the i^{th} influence, with $[1, 1, \cdots, 1]$ being a $(2 \times d_f)$-dimensional vector where all elements are 1, and $\parallel$ denotes the concatenation operator. The vector $[1, 1, \cdots, 1]$ can be regarded as a weight vector for every element in the connected feature vector $\vec{f}_i^{j\top} \parallel \vec{f}_i^{q\top}$, and it also can be set with various elements according to the specific situation. The attention index is calculated by Eq. (3) with both a_j's feature and its neighbors. Following this, the attention indexes for a_j are normalized to derive the initial link strength β^i_{jq} for each neighbor a_q under the i^{th} influence, as detailed in Eq. (4):

$$\beta^i_{jq} = \frac{exp(LeakyReLU(\alpha^i_{jq}))}{\sum_{q' \in N_{a_j}} exp(LeakyReLU(\alpha^i_{jq'}))} \tag{4}$$

This equation serves as a normalization mechanism for all attention indexes under each type of influence, setting the initial link strength from a_j to a_q for the i^{th} influence as $ls_i(a_j, a_q) = \beta^i_{jq}$. The comprehensive procedure is elaborated in Algorithm 1.

Algorithm 1 Initialization of link strength for MAS model

```
Input: A social network $G = \{V, E\}$, every agent's behavioral matrix $F^j$.
for i=1 to i=m do // $i^{th}$ influence
    for j=1 to j=$|V|$ do // agent $a_j$
        for every $q$ $(q \in N_{a_j})$ do //neighbors of $a_j$
            agent.AttentionIndex($\vec{f}_1^{j}$,$\vec{f}_1^{q}$) through Eq.(3);
        end for
        for every $q$ $(q \in N_{a_j})$ do
            if $Exist(edge(a_j, a_q))$ then
                $ls_i(a_j, a_q)$=agent.Normalization($\alpha^i_{jq}$);
            end if
        end for
    end for
end for
```

Link strengths, initially set based on the behavioral features $\vec{f}_1^{j}, \vec{f}_2^{j}, \cdots, \vec{f}_m^{j}$ of agent a_j, are dynamically updated by the agents themselves, reflecting observations of their neighbors' behaviors. The process for dynamically updating link strength is outlined in Algorithm 2. At each time step, all activated agents update their link strengths prior to the influence diffusion interactions, as specified in line 4. The state set ST^j of agent a_j (as defined in **User/Agent**) holds the information on whether and by which type of influence a_j has been activated, with the influence type's identification number i' being retrievable from IN_j. To be specific, if agent a_j has been activated by at least one influence, the identification numbers of influences that have successfully activated a_j are collected in IN_j through ST^j. Then, the link strengths for a_j under the influence type $i'^{th} (i' \in IN_j)$ are then updated based on a comparison of the corresponding features of a_j's neighbors, as detailed in Eq. (5):

$$ls_{i'}(a_j, a_q) = \begin{cases} p\ (p \sim U[0, \beta^{i'}_{jq}]), & \text{if } ED(\vec{f}^{j}_{i'}, \vec{f}^{q}_{i'}) \leq 0.5 \\ p'\ (p' \sim U[\beta^{i'}_{jq}, 1]), & \text{otherwise} \end{cases} \tag{5}$$

where $ED(\vec{f}^{j}_{i'}, \vec{f}^{q}_{i'})$ represents the Euclidean distance between the two behavioral feature vectors and is regarded a behavioral similarity (under $i'th$ influence) between a_j and $a_q (q \in N_{a_j})$. If the behavioral similarity is less than 0.5, the updated value follows a uniform distribution in the range of $[0, \beta^{i'}_{jq}]$. If the similarity is not lower than 0.5, the value follows a uniform distribution within the interval $[\beta^{i'}_{jq}, 1]$.

Algorithm 2 Dynamic link strength in MAS

```
Input: A social network $G = \{V, E\}$, agent set $A$ in MAS with initialized link strength matrix of every agent.
1: for $t = 1$ to $t = t_{max}$ do //$t^{th}$ time step
2:     for every agent $a_j \in A$ do
3:         if $sum(ST^j) \geq 1$ then //if it is an activated node
4:             Update($a_j$,$IN_j \leftarrow ST^j$); //dynamically updating the corresponding link strength according to Eq.(5)
5:         end if
6:     end for
7: end for
```

C. Agent interaction and MAS influence diffusion

In the MAS model, the process of influence diffusion is facilitated by interactions between agents, as depicted in Algorithm 3. On the one hand, the diffusion activity for an agent a_j, which was successfully activated in the preceding time step, is detailed from lines 3 to 9 of the algorithm. On the other hand, agent a_j is susceptible to activation through its activated neighbor agents, adhering to the predefined rules of the chosen diffusion model, as indicated from lines 10 to 18. The state of a_j is then adjusted in accordance with the outcomes of this diffusion process, as outlined in line 19.

In the context of the fixed diffusion model, during the diffusion phase for agent a_j, the set DN is defined to comprise the neighbors to whom a_j can extend its influence, as specified in line 4. a_j then proceeds to activate each neighboring agent in DN using the relevant influences, guided by the updated link strengths storing in LS^j. After the diffusion attempt, a_j

Algorithm 3 Agent interaction algorithm

```
1:  for every $a_j \in A$ do
2:      for t=1 to $t = t_{max}$ do
3:          if IsNewlyActivated($a_j$) then
4:              $DN \leftarrow$ DiffusionNeighbor($a_j$);
    //neighbors could be activated by $a_j$
5:              $LS^j \leftarrow$ UpdatedLinkStrength($a_j$);
    //get the updated value of link strength
6:              for every $a_{j'} \in DN$ do
7:                  Diffusing($a_j$,$a_{j'}$);
    //diffusing influence from $a_j$ to $a_{j'}$
8:              end for
9:          end if
10:         if ActivatedNeighbor($a_j$) then
11:             $AN \leftarrow$ ActivatedNeighbor($a_j$);
    //add to activated neighbor set of $a_j$
12:             for every $a_q \in AN$ do
13:                 if (DiffusionCapacity($a_q$)) then
    //if $a_q$ can activated $a_j$
14:                     $BDN \leftarrow a_q$;
    //add to BeDiffusedNodeSet of $a_j$
15:                     $LS^q \leftarrow$ UpdatedLinkStrength($a_q$);
16:                 end if
17:             end for
18:             BeDiffused($a_j$,$BDN$,ModelType);
    //$a_j$ accepts diffusion from neighbors
19:             UpdateState($a_j$);
20:         end if
21:     end for
22: end for
```

Algorithm 4 Multiple influences diffusion in MAS

```
Input 2: the seed set $S = s_1, s_2, \cdots, s_m$.
Initialization: algorithm 1();
for i=1 to i=m do
    $r_i = 0$;
end for
ActivatingSeedSet(S);
while NewlyActivated() do
    Dynamic link strength: algorithm 2();
    Agent interaction: algorithm 3();
    for i=1 to i=m do
        Records($r_i$);
    end for
end while
```

may itself become activated by any of its activated neighbors that possess the capability to diffuse influence, chosen from the set of already activated neighbors AN, as detailed in lines 13-14. Specially, the function in line 13 is a boolean function to determine whether the agent has diffusion capacity. This function works by checking activated agents: if the agent was successfully activated by the influence(s) in the last time step but not in the step before that, it possesses diffusion capability; otherwise, it does not. Then, all activated neighbors having diffusion capability are collected in the set BDN and a_j will accept the corresponding influences diffused by these neighbors under the fixed diffusion model, line 18. The state of a_j is updated to reflect the results of these diffusion attempts, as noted in line 19.

The setup for MAS in MIMDLS is outlined in Algorithms 1-3, covering initialization and dynamic updates of link strengths and agent interactions. The primary algorithm for multiple influences diffusion in MAS is detailed in Algorithm 4. A key feature is the tracking of non-seed activated nodes by each influence type, recorded in r_i, which also serves as a reward metric in Algorithm 5.

D. Distributed deep reinforcement learning for MIMDLS

Drawing inspiration from the asynchronous advantage actor-critic (A3C) algorithm [36], a distributed DRL framework is proposed for identifying optimal seed sets in MIMDLS problems, as depicted in Fig. 6. This framework features a mini MAS with m executor agents, where each agent $(agent_1, agent_2, \cdots, agent_m)$ is tasked with selecting seed nodes to maximize the spread of the $1^{st}, 2^{nd}, \cdots, m^{th}$ influence, respectively. These m agents function as distributed, though interconnected, threads within the DRL system, each exploring seed nodes within the social network (G).

To ensure that the seed sets for different influences remain mutually exclusive $(S_1 \cap S_2 \cap \cdots \cap S_m = \varnothing)$, the agents communicate with each other in each iteration, as indicated by the horizontal arrows in Fig. 6. Each agent operates its own Actor-Critic (AC) network, which is used to evaluate Q-values for potential diffusion outcomes initiated by selected seed nodes. The AC network also aids in aggregating gradient updates for the loss function following each agent's interaction with the environment G. Periodically, the agents update the parameters of a global AC network with their accumulated gradients and synchronize their local AC networks with this global network to align for subsequent iterations.

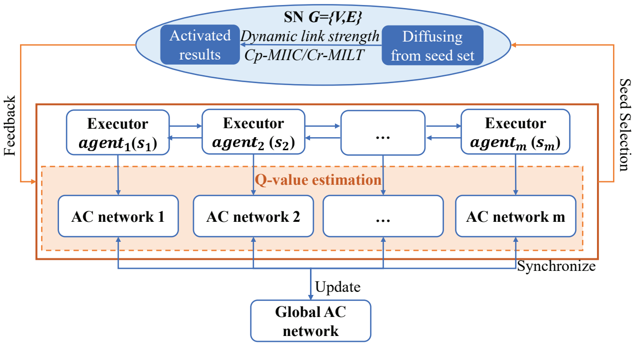


Fig. 6. A distributed DRL framework for MIMDLS

In the distributed DRL framework for solving MIMDLS problems, the essential components, namely, *action*, *reward*, *state*, and *policy*, are defined as follows:

1) State: The global state of the social network G at time step t, denoted as $State_t$, encapsulates the comprehensive set of current seed nodes. For each agent $agent_z$ dedicated to maximizing the z^{th} influence, its thread state,

represented as $state_t^{a_z}$, includes the specific seed nodes identified for that influence at time t.

2) Action: The action taken by $agent_z$ involves appending a new node to its designated seed set for the z^{th} influence. This selection is informed by a policy π, which dictates the choice of node.
3) Reward: The reward r^{a_z} resulting from an action by $agent_z$, i.e., the addition of a node to a seed set, is quantified by the number of nodes newly activated as a consequence of this action.
4) Policy: The policy π employed in this context adheres to a greedy approach, wherein nodes are chosen based on their potential to yield the highest Q-value, indicative of their expected contribution to influence diffusion.

These components collectively drive the decision-making process in the DRL algorithm, facilitating the dynamic and strategic selection of seed nodes to optimize influence spread within the network.

The distributed DRL algorithm for addressing the MIMDLS challenge is outlined in Algorithm 5. Here, the m executor agents represent individual threads interacting with the social network G at each time step. In line 6 of the algorithm, agents are tasked with performing actions (specifically, adding a node to their respective seed sets) while ensuring that the seed sets remain mutually exclusive, as indicated by $S_1 \cap S_2 \cap \cdots \cap S_m = \varnothing$. This mutual exclusivity prevents any executor agent from selecting a node already chosen by another, necessitating inter-agent communication to resolve any selection conflicts over a node $a_{j'}$. During such communications, agents assess the Q-value of $a_{j'}$ under their respective influences. The agent whose influence yields the highest Q-value for $a_{j'}$ gains the privilege to incorporate this node into its seed set. The other agents, in turn, must then opt for the node with the next highest Q-value as determined by their AC networks. Following the execution of their actions, the agents transition to their subsequent states and are rewarded based on the number of nodes activated as a result of the diffusion process within G.

The iterative cycle within the distributed DRL algorithm for the MIMDLS problem, spanning from line 6 to line 23, continues unless $agent_z$ reaches a terminal state or the designated time step pn_z is met. An agent enters a terminal state under specific circumstances: if there are no more viable candidate nodes left in the network G for selection ($CN() = 0$), if the global seed set is fully populated with k seeds in the Competitive MIMDLS (Cp-MIIC) scenario, or if the agent's individual seed set in the Cooperative MIMDLS (Cr-MILT) scenario reaches its limit of k/m seeds, as outlined in lines 13 to 15. Upon reaching the terminal state or at the pn_z time step, the agent ceases state transitions. The Q-value for the terminal state or the state at time step pn_z is either calculated or set to zero, detailed from lines 16 to 21. The process of computing Q-values and accumulating gradients for optimization occurs at each time step within the interval $[1, t-1]$, where t denotes the current time step, as described in lines 25-29. During this process, $r_b^{a_z}$ indicates the the reward of selecting a node (by agent a_z) in the b^{th} time step and its value equals to the number of nodes that are newly activated as a result of the selected node's diffusion. The concluding action of an episode involves synchronizing the agent-specific parameters with the global AC network, see line 30.

Algorithm 5 Distributed DRL algorithm for MIMDLS

Input 1: A social network $G = \{V, E\}$, seed number *k*, every agent's behavioral matrix F^j.

Input 2: the parameters of the global AC network: δ_g, ω_g, the global time step *T*, period time of every executor agent pn_z.

```
1:  for z=1 to z=m do
2:      $d\delta \leftarrow 0$, $d\omega \leftarrow 0$;  //Reset gradient of AC network z.
3:      $\delta' \leftarrow \delta_g$, $\omega' \leftarrow \omega_g$;  //Synchronizing, local parameters are $\delta', \omega'$
4:      $t = 1$;  //Reset iteration number of executor $agent_z$.
5:      $s_z = \varnothing$; $state^{a_z} = s_z$  //Initialize seed set
6:      if !SelectionConflict() then
7:          $action^{a_z} \xleftarrow{\pi} action_t^{a_z}$;
8:      else
9:          $action_t^{a_z} \leftarrow Compare(action_t^{a_z})$;
10:     end if
11:     $state^{a_z} \leftarrow state_{t+1}^{a_z}$, $r^{a_z} \xleftarrow{alg4} r_t^{a_z}()$;
12:     $t = t + 1$,$T = T + 1$;
13:     if Cp-MIIC?$|S| == k$:$|s_z| == k/m||$CN()==0 then
14:         $state^{a_z} \leftarrow state_{terminal}$
15:     end if
16:     if $t == pn_z || state^{a_z} == state_{terminal}$ then
17:         if $t == pn_z$ then
18:             $Q(state^{a_z}, t) = V(state^{a_z}(s_z), \omega')$
19:         else
20:             $Q(state^{a_z}, t) = 0$
21:         end if
22:     else
23:         go to line 6;
24:     end if
25:     for  b=t-1 to b=1 do
26:         $Q(state_b^{a_z}, b) = r_b^{a_z} + \gamma Q(state_{b+1}^{a_z}, b + 1)$;
27:         $d\delta \leftarrow d\delta + \bigtriangledown_{\delta'} \log \pi_{\delta'}(state_b^{a_z}, action_b^{a_z})$
                    $Q(state_b^{a_z}, action_b^{a_z}, b)$
                    $+ \bigtriangledown_{\delta'} H(\pi(state_b^{a_z}; \delta'))$;
28:         $d\omega \leftarrow d\omega + \frac{\partial(Q(state_b^{a_z}, b) - V(State_b, \omega'))^2}{\partial \omega'}$;
29:     end for
30:     Update($\delta_g, d\delta$), Update($\omega_g, d\omega$);  //Global updating
31:     if $T < T_{max}$ then
32:         go to line 2;
33:     else
34:         Final parameters: $\delta_g$ and $\omega_g$.
35:     end if
36: end for
```

A key feature distinguishing the DRL approach for the MIMDLS problem is the interconnected nature of the m threads, representing the m agents. Unlike completely independent threads, these agents engage in communication to prevent overlap in their seed node selections, ensuring that their actions are coordinated and conflict-free. Moreover, the simultaneous action-taking by the m agents on the social network G introduces a layer of complexity where the reward for an action taken by one agent is not isolated but can be affected by the concurrent actions of the other agents. This interdependency means that the outcome of any single action is a result of not just the individual agent's strategy but also the collective dynamics of all agents' decisions within the network at that time.

V. Experiments and Analysis

This section delves into the experimental setup, including the environment, database, methods for comparison, and the results obtained from these tests. The specifics regarding the database and parameter configurations for the MIMDLS experiments, along with the comparison methods, are detailed in Section V-A. Following this, Section V-B evaluates the effectiveness and efficiency of the proposed methods within the context of the Cp-MIIC model and the Cr-MILT model, across a variety of experimental scenarios. The comparative performance analysis, juxtaposing the proposed methods against both baseline and cutting-edge techniques, is provided in Section V-C. The implementation of the proposed algorithms was carried out using C++ and Python programming languages and executed on a CPU-based computing infrastructure.

A. Database, parameters settings and compared methods

1) Database: The experimental database utilizes real-world network data from SinaWeibo, gathered using Python crawlers, encompassing interactions among 5000 users. This dataset not only includes user connections but also tracks four behavioral activities (commenting, liking, posting, and mentioning) for each user across a span of 400 days. The four behaviors are indicated as b_1 (commenting), b_2 (liking), b_3 (posting) and b_4 (mentioning) in the following experiments. These interactions form the basis for constructing the social network, with user links defining relationships and behavior logs contributing to feature vector creation, each with a dimensionality of 4, refined through data preprocessing.

For the purpose of these experiments, the dataset is structured to facilitate the study of diffusion dynamics across five distinct types of influences. Consequently, the preprocessing phase involves segmenting each user's behavioral data into five subsets, corresponding to the different influences, as detailed in Algorithm 6. The main steps are as follows. **Step 1**: As outlined from lines 2 to 14, the behavioral data collected over 400 days is divided into five distinct sets labeled $behavior_{x'}$, where x' ranges from 1 to 5, each set representing behaviors under a specific type of influence. (The behaviors that occurred on the x^{th} day are stored in $recordsDay(x)$.) **Step 2**: The probability of each behavior type occurring is computed by dividing the count of a specific behavior by the aggregate count of all behaviors, thereby normalizing the data. See lines 15-16, where $|b_1|$ indicates the occurrence number of b_1 behavior. **Step 3**: With these probabilities computed, the feature vector $\vec{f_1^j}, \vec{f_2^j}, \vec{f_3^j}, \vec{f_4^j}$ and $\vec{f_5^j}$ for each influence type are readily established, as indicated in line 17.

Algorithm 6 Data preprocessing

```
1:  for every user (regard as agent) $a_j$ in database do
2:      for x=1 to x=400 do
3:          if x%5==0 then
4:              $behavior_1 \leftarrow recordsDay(x)$;
5:          else if x%5==1 then
6:              $behavior_2 \leftarrow recordsDay(x)$;
7:          else if x%5==2 then
8:              $behavior_3 \leftarrow recordsDay(x)$;
9:          else if x%5==3 then
10:             $behavior_4 \leftarrow recordsDay(x)$;
11:         else
12:             $behavior_5 \leftarrow recordsDay(x)$;
13:         end if
14:     end for
15:     for $x' = 1$ to $x' = 5$ do
16:         $p_1 = |b_1|/|behavior_{x'}|$, $p_2 = |b_2|/|behavior_{x'}|$, $p_3 = |b_3|/|behavior_{x'}|$, $p_4 = |b_4|/|behavior_{x'}|$;
17:         $f_{x'}^j = \{p_1, p_2, p_3, p_4\}$
18:     end for
19: end for
```

It should be mentioned that there is no data for threshold vector θ^j ($\theta^j = \{\theta_0^j, \theta_1^j, \cdots, \theta_m^j\}$) of each agent a_j under Cr-MILT model, but thresholds can be initialised through original link strength generated via Algorithm 1. The value of θ^j for a_j is calculated by following equations:

$$\theta_i^j = Avg(\frac{ls_i(a_j, a_q)}{\sum_{\rho=1}^{m} ls_\rho(a_j, a_q)}) \quad (i = 1, 2, \cdots, m. \quad q \in N_{a_j}) \tag{6}$$

It can be seen that the value of θ_i^j is the average link strength of the i-th influence across all out-degrees of a_j.

$$\theta_0^j = \textbf{count}((\sum_{q=1}^{|N_{a_j}|} ls_i(a_q, a_j) \geq 0.5)?\ 1 : 0) \tag{7}$$

where $i = 1, 2, \cdots, m$, $q \in N_{a_j}$, and **count**() is a count function, when the summation of link strength of the i-th influence across all in-degrees of a_j exceeds 0.5, the function will count 1; otherwise it will count 0.

2) Parameter settings: For the experiments involving diffusion of multiple influences, in line with the database setup described earlier, five distinct types of influence are considered, with each user's behavior represented by a 4-dimensional feature vector. The parameters for the MIMDLS experiments are configured as follows. 1) The number of influence types m is set to 5, aligning with the number of distinct influences being studied. 2) The seed set S is structured as a collection of subsets $\{S_1, S_2, S_3, S_4, S_5\}$, each corresponding to one of the five types of influence. 3) The feature matrix for each agent F^j comprises five feature vectors $\vec{f_1^j}, \vec{f_2^j}, \vec{f_3^j}, \vec{f_4^j}, \vec{f_5^j}$, reflecting the user's behaviors under each influence type; the 5 feature vectors are generated through data preprocessing in Algorithm 6. 4) The dimensionality d_f of the 5 feature vectors $(\vec{f_1^j}, \vec{f_2^j}, \cdots, \vec{f_5^j})$ of one user is set to 4, consistent with the number of behavioral metrics considered.

In the distributed DRL framework tailored for the MIMDLS scenario, there are five executor agents within the miniature MAS, each responsible for optimizing the selection of seed nodes for one of the influence types. The discount factor γ utilized in the DRL algorithm is set to 0.95, indicating the degree to which future rewards are considered in the current value estimation. The dimension of the state for each agent $agent_z$ in the DRL model varies based on the diffusion model being applied: it is k for the Cp-MIIC model, indicating the total number of seed nodes across all influences, and $k/5$ for the Cr-MILT model, reflecting an equitable distribution of seed nodes among the five types of influence.

Algorithm 4, embodying the core logic of influence diffusion within this framework, has undergone training using the SinaWeibo dataset, with experiments conducted across various scales of the network, including subsets of 500, 1000, 2000, 3000, and 5000 users. This progressive approach enables an evaluation of the algorithm's performance and scalability in relation to network size.

3) Comparison methods: In the comparison experiments for the MIMDLS problem, several approaches are evaluated alongside the proposed method.

GE-based method [30]: This approach utilizes Graph Embedding (GE) to represent each node's features, incorporating neighborhood topology into a new vector. The potential for a node to spread influence is estimated using the GE-based method, with subsequent aggregation of feature information and estimation to refine the embedding vector and guide seed selection for IM. Since the GE-based method is not directly applicable to MIM, MIM is decomposed into separate IM problems, with the ultimate seed selection drawn from the aggregated solutions of these IM instances.

Max-k coverage [3]: Employing a concept akin to reverse influence sampling, this method begins by generating several reverse reachable sets. It then addresses the maximum k coverage problem across these sets by dynamically updating the incremental value to identify the most influential nodes.

Basic Greedy [25]: Serving as a foundational approach, this method selects nodes based on their maximum influence spread. For MIM challenges, it chooses one node per influence type in each iteration, ensuring an even distribution of seed nodes across all influence types.

Community-based [42]: This method assesses the influence potential of community-centric nodes to identify key influencers. Seed candidate sets are determined heuristically at the community level, with the final seed set compiled from these preliminary selections.

Play-strategy method [32]: Incorporating game-theoretic principles, this strategy treats each influence type as a player in a competitive setting. Seed node selection is informed by anticipating and countering the strategies of other influences, aiming for an optimal response in a competitive influence diffusion scenario.

CoreQ [2]: Relying on K-core hierarchies to divide the network topologies and guide the identification of seed nodes, this approach optimizes the selection of seed nodes through a Q-learning algorithm.

MIM-Reasoner [14]: Combining RL with a probabilistic graphical model, the MIM-Reasoner analyzes the complex diffusion within a layered network and optimizes the seed set for each layer through RL methods. To address the MIM problem, SN can be divided as multiple layers according to the number of influences.

MAS-based DRL method: a MAS-based DRL method has a single AC network compared to the proposed methods. In this method, only one agent is set for deep reinforcement learning, which exchanges information with the MAS environment (simulated by the proposed models). Thus, this agent set in DRL will select seed nodes for all influence.

B. Experiments on various settings for proposed models

The evaluation of the proposed models and algorithms is conducted with user populations of 500, 1000, 2000, 3000, and 5000 from the SinaWeibo dataset, employing snowball sampling for network extraction. Seed set sizes are varied among 10, 25, 50, 75, and 100 to test the models Cp-MIIC and Cr-MILT. Results are averaged over 100 runs and presented in Fig. 7 for Cp-MIIC and Fig. 8 for Cr-MILT, showcasing the diffusion effectiveness under different settings.

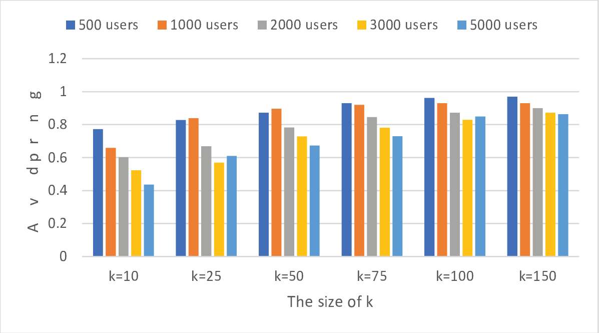


Fig. 7. The percentages of activated nodes in different settings under Cp-MIIC.

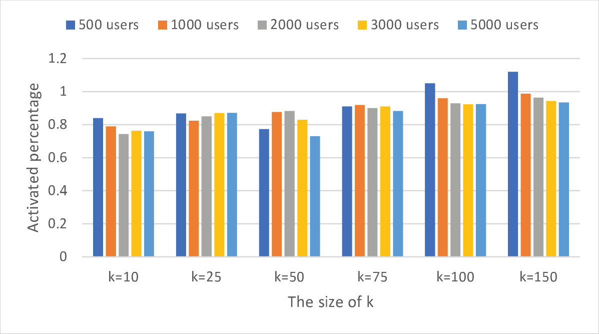


Fig. 8. The percentages of activated nodes in different settings under Cr-MILT.

In Fig. 7, the average diffusion percentages under Cp-MIIC with various settings are provided. Notably, the poorest performance occurs when k=10, which is attributed to the relatively low percentage of the seed set. For an agent population of 5000, achieving a diffusion percentage of 43% corresponds to a seed set size of k%=0.02%(k=100). Conversely, in a network of 500 agents, the diffusion percentage exceeds 85% when k>50, and nearly all nodes are activated when k=150 due to k%=0.3%. The proposed methods yield satisfactory performance when k%>1.5%.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TNNLS.2025.3588236, IEEE Transactions on Neural Networks and Learning Systems

The results under the Cr-MILT model are displayed in Fig. 8. A distinct difference from Fig. 7 is the higher diffusion percentages. This phenomenon is caused by the fact that activated nodes are counted based on the number of successful activations. Notably, when the population size is 500 and $k \geq 100$, the average diffusion percentage surpasses 1, implying that numerous nodes have been activated multiple times.

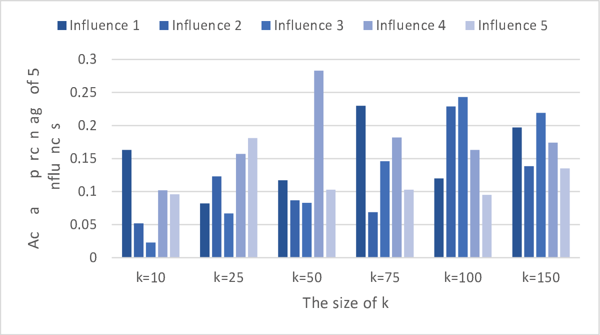


Fig. 9. The performance of five influences under Cp-MIIC.

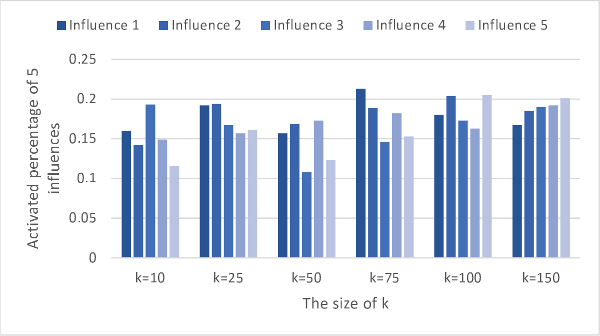


Fig. 10. The performance of five influences under Cr-MILT.

The performance of the five types of influence under the two models is depicted in Fig. 9 and Fig. 10, with the population setting at 5000. In Fig. 9, it is easy to find that the percentages of diffusion for the five influences fluctuate significantly compared to Fig. 10. A competitive result is reflected by Fig. 9 because one node only can be activated once in the spread of Cp-MIIC.

The other four networks (Facebook, Campus Forums, Flickr and Epinions) are taken into consideration to verify the effectiveness of the proposed method under 5 influences diffusion. The related properties of these networks are displayed in Table I. All links among the nodes of the four networks are collected in the database, along with their behaviors described in Table I, except for the Facebook network. Especially, Facebook database has 4,039 nodes, each with its links and a 40-dimensional feature. The feature is divided into five 8-dimensional vectors (indicating $\vec{f}_1^j, \vec{f}_2^j, \vec{f}_3^j, \vec{f}_4^j$ and $\vec{f}_5^j$) for multiple influence diffusion (m=5). Thus, the feature matrix of each node for five (m=5) influences is included in the Facebook database and does not need to be calculated.

The performance of five networks under two diffusion models are provided in Fig. 11. When k>150 under the 4 networks (except for Epinions), it can be seen that the percentage remains at a high level, ranging from 75% to 89% under Cp-MIIC model, and from 85% to 95% under Cr-MILT model. Especially, when k reaches 200 under the Epinions network, the percentage of activated nodes almost reaches 70% under Cp-MIIC model. When $k \geq 150$ under Cr-MILT model, the percentage ranges from 70% to 82%. The reason may be that the large scale of the network requires more seed nodes to obtain a percentage more than 80%.

TABLE I
PROPERTIES OF 5 NETWORKS.

Network	d_f	Behaviors	Size
Facebook	8	-	4039
CampusForums	5	commenting,liking,posting,grouping	3500
Flickr	3	sharing,commenting,grouping	8721
Epinions	3	reviewing,trusting,distrusting	22166
SinaWeibo	4	commenting,liking,posting,mentioning	5000

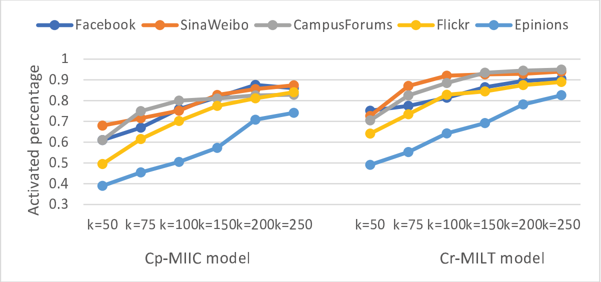


Fig. 11. The performance on various networks under the two models.

C. Comparison with advanced methods

Several advanced methods are compared with the proposed method, the distributed DRL algorithm, using a population size of 5000 users and diffusing five influences under the two diffusion models. After conducting each approach 100 times, the average performance is presented in Fig. 12 and Fig. 13.

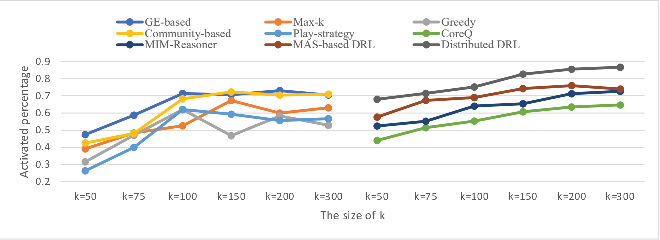


Fig. 12. Comparison of various methods under Cp-MIIC model.

Comparing the results of the Cp-MIIC model, the worst performance is observed from the play-strategy and greedy approaches in Fig. 12. In the Cp-MIIC model, the competitive relationship among influences can be managed through the play-strategy method, but the dynamics of MIMDSL cannot be resolved. With the increasingly growing seed set, a decreasing diffusion range is displayed by the play-strategy method when $k \geq 150$. Furthermore, anticipating and countering in the play simulation become complicated due to dynamic LS, thus the precision of the calculation for play profits has been affected. In the greedy approach optimal seed node computed in the current step may perform worse in the next step after influences diffusion, especially after significant changes have occurred among the LS. A waving line is presented by the greedy approach because the topology structure is concentrated by greedy idea but the competition among users and influences are ignored. However, the calculation precision of topology structure is compromised by dynamic link strength, which leads to an unstable result.

Although GE-based algorithm focuses on topology structure, the information embedding and aggregation help to improve diffusion to around 76%. The result of GE-based algorithm reflects the point that the effect of dynamic LS can be partially tackled through the method of information embedding and aggregation, although the effect is not satisfied. Better results are provided by the MIM-Reasoner than by the GE-based algorithm because layers corresponding to the respective influence are tackled simultaneously by the MIM-Reasoner. Worse performance is obtained by the CoreQ approach, but it is still higher than that of the max-k and community-based methods. The reason is the optimization of seed selection through Q-learning algorithm in CoreQ. The MIMDLS is divided into several sub-graphs by max-k and community-based approaches, and the diffusion percentage is between 70% and 80% when $k \geq 150$. It can be seen that within asynchronous training in the AC networks, dynamic LS can be handled by Distributed DRL framework skillfully. The Q value of optimal seed nodes is estimated precisely by the trained global AC network in every step, adapting the corresponding influence diffusion simulated by agent interactions. The best results are shown by the distributed DRL framework with an increased tendency of a green line and the maximum diffusion is about 86%. Both competitive relationship and dynamics are taken into consideration and a stable diffusion solution is generated whatever k. Compared with the distributed DRL framework, lower diffusion percentages are obtained by MAS-based DRL. A single AC network has the disadvantage that the training model sometimes is hard to ensure convergence. This is the reason why MAS-based DRL's performance is worse than distributed DRL.

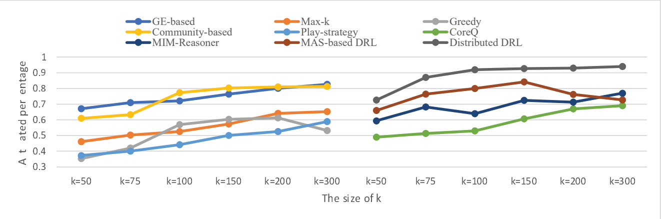


Fig. 13. Comparison of various methods under Cr-MILT model.

The corresponding results for Cr-MILT are depicted in Fig. 13. All performance outcomes are relatively stable due to the cooperative relationships among influences and the condition that each node can be activated multiple times. The results of play-strategy algorithm are due to the excessive calculations focused on agent benefits in play simulation. The improvement of cooperative MIM is gently hindered by competitive play strategies. However, higher percentages are generated by the CoreQ method, which benefits from Q-learning optimization. The greedy method is more effective under the Cr-MILT model than under the Cp-MIIC model, because the fairness of cooperative influences is ensured by the greedy idea. Although relatively higher results are provided by the MIM-Reasoner approach, it still cannot exceed 80% because the layered calculation cannot handle the dynamic LS. The GE-based method is more efficient than the MIM-Reasoner, which reflects that the representation of the graph structure is more appropriate than the layered algorithm under the Cr-MILT model. Benefiting from the loose constraints of Cr-MILT, a middle position is occupied by the max-k idea with an orange line. But the dynamics lead to a decrease tendency when k>150. With the increasing number of *k*, more subsets of the seed set will be calculated along with their corresponding incremental values via the max-k algorithm. However, every calculation suffers from dynamic LS, which decreases the positive effect of incremental value. MAS-based DRL excesses GE-based and community-based approaches when k<200, because the dynamic LS can be tackled well by single AC network. Besides, this situation has not be remained until $k \geq 200$, because large iterations cannot ensure parameter convergence to an optimal value, leading to a worse quality of seed set. In contrast, the proposed distributed DRL method maintains a stable performance, achieving an activation range of 85% to 95% of nodes.

Next, the performance of various technologies is analyzed based on the diffusion speed. The diffusion speed is reflected by the iteration length, which refers to the number of diffusion iterations from the activation of seed nodes until no further nodes can be activated. It should be noted that the iteration length is recorded by simulating the diffusion of multiple influences in MAS, based on the seed set generated by each approach, respectively. The longer the iteration length, the slower the diffusion speed. Each method is run to generate the optimal seed set and then multiple influence diffusion is simulated according to the seed set to obtain the iteration length. These steps are repeated 100 times, and the results are shown in Table II. For instance, based on the 100 results of the GE-based algorithm under Cp-MIIC model, after 100-time simulation of diffusion, the average iteration length is 52, the maximum is 84, and 23 is the minimum value.

Under Cp-MIIC model, max-k, CoreQ and basic greedy techniques have shorter average iteration lengths (33, 33 and 34) than others, meaning that diffusion is highly likely to complete at a fast speed according to their optimal seed sets. However, max-k and greedy methods also have the lower diffusion percentages than others in Fig. 12 and Fig. 13. The GE-based and Play-strategy methods have longer average iteration lengths (52 and 44), while the remaining approaches have medium average iteration lengths (distributed DRL ranks sixth). The slowest speed is caused by the community-based approach with an iteration length 91, while the fastest is with max-k at 16. Besides, the shortest performance of the proposed method is 21. The GE-based, MIM-Reasoner and MAS-based DRL methods are the closest to the proposed

Citation information: DOI: 10.1109/TNNLS.2025.3588236, IEEE Transactions on Neural Networks and Learning Systems

TABLE II
THE ITERATION LENGTH OF INFLUENCE DIFFUSION OF SOLUTIONS FROM VARIOUS METHODS.

	k=150 under Cp-MIIC		
Algorithm	Average iteration	Maximum	Minimum
GE-based	52	84	23
Max-k	33	57	16
Basic Greedy	34	72	19
Community-based	39	91	24
Play-strategy	44	75	26
CoreQ	33	54	25
MIM-Reasoner	43	67	23
MAS-based DRL	37	82	24
Distributed DRL	42	79	21
	k=150 under Cr-MILT		
GE-based	67	124	39
Max-k	72	121	42
Basic Greedy	49	102	20
Community-based	68	113	41
Play-strategy	52	95	37
CoreQ	65	89	46
MIM-Reasoner	62	97	44
MAS-based DRL	55	104	40
Distributed DRL	51	107	43

method in terms of diffusion percentage under the Cp-MIIC model, and they exhibit similar performance in terms of maximum and minimum iteration length. Under the Cr-MILT model, CoreQ's performance is similar to MIM-Reasoner's. Besides, MAS-based DRL and play-strategy methods have the closest performance to the proposed method in terms of diffusion percentage, but they have longer average iteration lengths compared to distributed DRL (which ranks second). The longest performance is 124, generated by the GE-based method, and the shortest is 20, generated by the basic greedy method. The shortest iteration length of distributed DRL is 43.

TABLE III
RUNNING TIME (MIN) FROM VARIOUS METHODS UNDER 5 INFLUENCES ($k = 150$).

Approach \ User population	1000	2000	3000	5000
GE-based	15	21	32	128
Max-k	11	14	17	58
Basic Greedy	145	257	$>10^3$	$>10^3$
Community-based	151	181	846	$>10^3$
Play-strategy	33	166	370	764
CoreQ	13	25	39	67
MIM-Reasoner	16	25	41	63
MAS-based DRL	55	114	189	301
Distributed DRL	18	27	40	62

The time complexity analysis of the distributed DRL algorithm (Algorithm 4) can be estimated under two diffusion models. In the Cr-MIMDLS model, the worst-case scenario is that the Q-value of each non-seed node will be recomputed by the executor agent during every selection and no node reaches its threshold of influence number during the selection. Thus, the upper bound of the complexity is $O(k(2n-k+m)/2m)$. Besides, in Cr-MIMDLS model, the worst circumstances is that one executor agent selects k seed nodes and the Q-value of each non-seed node is updated in agent's every iteration. The resulting upper bound of the complexity is: $O(n+k(1-k)/2)$. In order to further evaluated the proposed methods, the running time is provided in this section.

Running time of varies methods under different population sizes are displayed in Table. III. Here, the size of seed set is 150 and each result is the average time after running the corresponding algorithm 30 times. The Max-k approach has a low running time in all situations, but a worse seed set is generated by this approach compared to other methods. The reason Max-k uses less time is that subsets of the current seed set constitute the search space, which accelerates the search speed and its speed has little help for influence diffusion. The CoreQ method has a slightly longer running time than Max-k because of the optimization steps. Similar performance is displayed by the MIM-Reasoner. The reason is that it can tackle network layers in parallel and conduct batch inference. The structural feature is extracted by the GE-based algorithm, which achieves a low running time when the population is no more than 3,000, but the running time increases significantly when the SN has 5,000 users. The worst performance is observed when applying the Basic Greedy method, where the time reaches 10^3 due to heavy calculations in each iteration of the complex influence diffusion. Running times of Community-based and Play-strategy approaches are slightly longer. Complicated community division due to dynamic LS and greedy calculation takes time in the Community-based algorithm. The convergence process becomes lengthy due to dynamic LS when the play-strategy method simulates play profits among influences. The reason that Distributed DRL takes less time than MAS-based DRL is that several AC networks for influences are executed in parallel in Distributed DRL, whereas only serial computation is performed on a single deep neural network in MAS-based DRL. By the way, the training time is significantly reduced by Distributed DRL, to no more than one-third of that of MAS-based DRL.

VI. CONCLUSION

This paper has presented a distributed DRL framework within an MAS to address the MIMDLS problem in social networks by utilizing Cp-MIIC and Cr-MILT models. The approach has leveraged user behavior vectors to dynamically model link strengths and devise an MAS-based diffusion strategy for multiple influences, incorporating specific interaction rules. The distributed nature of MAS has inspired the development of a DRL model for MIMDLS, which has been shown to be both effective and efficient. The model facilitates asynchronous updates by agents, who communicate to meet the requirements of different diffusion models. When compared to five advanced methods, our proposed algorithm has consistently outperformed others across a range of settings. Experimental outcomes have affirmed the proposed models and algorithms' capability to effectively tackle the MIMDLS problem. While the proposed approach has shown promise for MIM problems, its reliance on predefined node feature vectors is a noted limitation. Future research could explore MIM challenges where node information is partially known or evolving, expanding the applicability of these models and algorithms.

Citation information: DOI: 10.1109/TNNLS.2025.3588236, IEEE Transactions on Neural Networks and Learning Systems

References

[1] C. C. Aggarwal, S. Lin and P. S. Yu,“On influential node discovery in dynamic social networks,” in *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 636–647, 2012.

[2] W. Ahmad and B. Wang, “A learning-based influence maximization framework for complex networks via K-core hierarchies and reinforcement learning,” *Expert Systems with Applications*, vol. 259, pp. 125393, 2025.

[3] C. Borgs, M. Brautbar, J. T. Chayes and B. Lucier, “Maximizing social influence in nearly optimal time,” in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 946–957, 2014.

[4] A. Caliò and A. Tagarelli, “Attribute based diversification of seeds for targeted influence maximization,” *Information Sciences*, vol. 546, pp. 1273–1305, 2021.

[5] M. Charikar, Y. Naamad and A. Wirth, “On approximating target set selection,” *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, 2016.

[6] B. Chen, Y. Shen, M. Ji, J. Liu, Y. Yu and Y. Zhang, “Competitive Influence Maximization on Online Social Networks under Cost Constraint,” *KSII Transactions on Internet & Information Systems*, vol. 15, no. 4, 2021.

[7] H. Chen, W. Qiu, H. Ou, B. An and M. Tambe, “Contingency-aware influence maximization: A reinforcement learning approach,” *Uncertainty in Artificial Intelligence*, pp. 1535–1545, 2021.

[8] T. Chen, J. Guo and W. Wu,“Graph representation learning for popularity prediction problem: a survey,” *Discrete Mathematics, Algorithms and Applications*, vol. 17, no. 07, pp. 2230003, 2022.

[9] X. Chen, G. Song, X. He and K. Xie, “On influential nodes tracking in dynamic social networks,” in *Proceedings of the 2015 SIAM International Conference on Data Mining*, pp. 613–621, 2015.

[10] Y. Chen, W. Zhu, W. Peng, W. Lee and S. Lee, “Cim: Community-based influence maximization in social networks,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 2, pp. 1–31, 2014.

[11] E. Cohen, D. Delling, T. Pajor and R. F. Werneck, “Sketch-based influence maximization and computation: Scaling up with guarantees,” in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 629–638, 2014.

[12] Q. Cui, K. Liu, Z. Ji and W. Song, “Sampling-data-based distributed optimisation of second-order multi-agent systems with PI strategy, *International Journal of Systems Science*, vol. 54, no. 6, pp. 1299–1312, 2023.

[13] A. R. da Silva, R. F. Rodrigues, V. da Fonseca Vieira and C. R. Xavier, “Influence maximization in network by genetic algorithm on linear threshold model,” in *Computational Science and Its Applications–ICCSA 2018: 18th International Conference, Melbourne, VIC, Australia, July 2-5, 2018, Proceedings, Part I 18*, pp. 96–109, 2018.

[14] N. H. K. Do, T. Chowdhury, C. Ling, L. Zhao and M. T. Thai, “MIM-Reasoner: Learning with Theoretical Guarantees for Multiplex Influence Maximization,” in *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pp. 2296–2304, 2024.

[15] S.-S. Dong, Y.-G. Li and L. An, “Optimal strictly stealthy attacks in cyber-physical systems with multiple channels under the energy constraint”, *International Journal of Systems Science*, vol. 54, no. 13, pp. 2608–2625, 2023.

[16] P. Domingos and M. Richardson, “Mining the network value of customers,” in *Proceedings of the 7th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 57–66, 2001.

[17] J. Fang, W. Liu, L. Chen, S. Lauria, A. Miron and X. Liu, “A survey of algorithms, applications and trends for particle swarm optimization”, *International Journal of Network Dynamics and Intelligence*, vol. 2, no. 1, pp. 24–50, Mar. 2023.

[18] W. Fang, B. Shen, A. Pan, L. Zou and B. Song, “A cooperative stochastic configuration network based on differential evolutionary sparrow search algorithm for prediction”, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2314481, 2024.

[19] B. Fu, J. Zhang, H. Bai, Y. Yang and Y. He, “An influence maximization algorithm for dynamic social networks based on effective links,” *Entropy*, vol. 24, no. 7, pp. 904, 2022.

[20] S. Galhotra, A. Arora and S. Roy, “Holistic influence maximization: Combining scalability and efficiency with opinion-aware models,” in *Proceedings of the 2016 international conference on management of data*, pp. 743–758, 2016.

[21] S. Galhotra, A. Arora, S. Virinchi and S. Roy, “Asim: A scalable algorithm for influence maximization under the independent cascade model,” in *Proceedings of the 24th International Conference on World Wide Web*, pp. 35–36, 2015.

[22] P. Gao, C. Jia and A. Zhou, “Encryption-decryption-based state estimation for nonlinear complex networks subject to coupled perturbation,” *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2357796, 2024.

[23] H. Huang, Z. Meng and H. Shen, “Competitive and complementary influence maximization in social network: A follower’s perspective,” *Knowledge-Based Systems*, vol. 213, pp. 106600, 2021.

[24] F. Jin, L. Ma, C. Zhao and Q. Liu, “State estimation in networked control systems with a real-time transport protocol”, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2347885, 2024.

[25] D. Kempe, J. Kleinberg, É. Tardos, “Maximizing the spread of influence through a social network,” *Theory OF Computing*, vol. 11, pp. 105–147, 2015.

[26] M. Khajehnejad, A. A. Rezaei, M. Babaei, J. Hoffmann, M. Jalili and A. Weller, “Adversarial graph embeddings for fair influence maximization over social networks,” in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 4306–4312, 2021.

[27] C. Li, Y. Liu, M. Gao and L. Sheng, “Fault-tolerant formation consensus control for time-varying multi-agent systems with stochastic communication protocol”, *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 1, art. no. 100004, Mar. 2024.

[28] H. Li, M. Xu, S. S. Bhowmick, S. R. Joty, C. Sun and J. Cui,“PIANO: Influence maximization meets deep reinforcement learning,” *IEEE Transactions on Computational Social Systems*, vol. 10, no. 3, pp. 1288–1300, 2023.

[29] L. Li, Y. Liu, Q. Zhou, W. Yang and J. Yuan, “Targeted influence maximization under a multifactor-based information propagation model,” *Information Sciences*, vol. 519, pp. 124–140, 2020.

[30] M. Li, Z. Wang, Q. Han, S. J. E. Taylor, K. Li, X. Liao, X. Liu, “Influence maximization in multiagent systems by a graph embedding method: dealing with probabilistically unstable links,” *IEEE Transactions on Cybernetics*, vol. 53, no. 9, pp. 6004–6016, 2023.

[31] W. Li, Y. Hu, C. Jiang, S. Wu, Q. Bai and E. Lai, “ABEM: An adaptive agent-based evolutionary approach for influence maximization in dynamic social networks,” *Applied Soft Computing*, vol. 136, pp. 110062, 2023.

[32] S.-C. Lin, S.-D. Lin and M.-S. Chen, “A learning-based framework to handle multi-round multi-party influence maximization on social networks,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 695–704, 2015.

[33] Y. Li, H. Gao, Y. Gao, J. Guo and W. Wu, “A survey on influence maximization: From an ml-based combinatorial optimization,” *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 9 pp. 1–50, 2023.

[34] I. Lozano-Osorio, J. Sánchez-Oro and A. Duarte,“A variable neighborhood search approach for the adaptive multi round influence maximization problem,” *Social Network Analysis and Mining*, vol. 14, no. 1, pp. 1–18, 2024.

[35] Y. Meng, Y. Yi, F. Xiong and C. Pei, “T× one Hop approach for dynamic influence maximization problem,” in *Physica A: Statistical Mechanics and its Applications*, vol. 515, pp. 575–586, 2019.

[36] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International conference on machine learning*, pp. 1928–1937, 2016.

[37] R. Narayanam and Y. Narahari, “A shapley value-based ap proach to discover influential nodes in social networks,” *IEEE Transactions on Automation Science and Engineering* , vol. 8, no. 1, pp. 130–147, 2010. H. T. Nguyen, P. Ghosh, M. L. Mayo and T. N. Dinh, “Social influence spectrum at scale: Near-optimal solutions for multiple budgets at once,” *ACM Transactions on Information Systems (TOIS)*, vol. 36, no. 2, pp. 1–26, 2017.

[38] S. Nishad, S. Agarwal, A. Bhattacharya and S. Ranu, “GraphReach: Position-Aware Graph Neural Network using Reachability Estimations,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1527–1533, 2021.

[39] N. Ohsaka, T. Akiba, Y. Yoshida and K. Kawarabayashi, “Dynamic influence analysis in evolving networks,” in *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1077–1088, 2016.

[40] G. Panagopoulos, F. D. Malliaros and M. Vazirgiannis, “Influence maximization using influence and susceptibility embeddings,” in *Proceedings*

Citation information: DOI: 10.1109/TNNLS.2025.3588236, IEEE Transactions on Neural Networks and Learning Systems

of the International AAAI Conference on Web and Social Media, vol. 14, pp. 511-521, 2020.

[41] X. Qin, C. Zhong, H. Lin, "Community-Based Influence Maximization Using Network Embedding in Dynamic Heterogeneous Social Networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 8, pp. 1–21, 2023.

[42] J. Shang, S. Zhou, X. Li, L. Liu and H. Wu, "CoFIM: A community-based framework for influence maximization on large-scale networks," *Knowledge-Based Systems*, vol. 117, pp. 88-100, 2017.

[43] Y. Shang, B. Zhou, Y. Wang, A. Li, K. Chen, Y. Song and C. Lin, "Popularity prediction of online contents via cascade graph and temporal information," *Axioms*, vol. 10, no. 3, pp. 159, 2021.

[44] S. S. Singh, K. Singh, A. Kumar and B. Biswas, "MIM2: Multiple influence maximization across multiple social networks,"*Physica A: Statistical Mechanics and its Applications*, vol. 526, pp. 120902, 2019.

[45] H. Sun, X. Gao, G. Chen, J. Gu and Y. Wang, "Multiple influence maximization in social networks," in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, pp. 1–8, 2016.

[46] Y. Sun, M. Chen, K. Peng, L. Wu and C. Liu, "Finite-time adaptive optimal control of uncertain strict-feedback nonlinear systems based on fuzzy observer and reinforcement learning," *International Journal of Systems Science*, vol. 55, no. 8, pp. 1553–1570, 2024.

[47] J. Tang, X. Tang and J. Yuan, "An efficient and effective hop-based approach for influence maximization in social networks," *Social Network Analysis and Mining*, vol. 8, pp. 1–19, 2018.

[48] L. Tian, "Multi-dimensional adaptive learning rate gradient descent optimization algorithm for network training in magneto-optical defect detection," *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 3, art. no. 100016, Sept. 2024.

[49] S. Tian, S. Mo, L. Wang and Z. Peng, "Deep reinforcement learning-based approach to tackle topic-aware influence maximization," *Data Science and Engineering*, vol. 5, pp. 1–11, 2020.

[50] J. Tong, L. Shi, L. Liu, J. Panneerselvam and Z. Han, "A novel influence maximization algorithm for a competitive environment based on social media data analytics," *Big Data Mining and Analytics*, vol. 5, no. 2, pp. 130–139, 2022.

[51] C. Tsai, Y. Yang, M. Chiang, "A genetic newgreedy algorithm for influence maximization in social network," *2015 IEEE international conference on systems, man, and cybernetics*, pp. 2549–2554, 2015.

[52] Y. Wang, W. Liu, C. Wang, F. Fadzil, S. Lauria and X. Liu, "A novel multi-objective optimization approach with flexible operation planning strategy for truck scheduling," *International Journal of Network Dynamics and Intelligence*, vol. 2, no. 2, art. no. 100002, Jun. 2023.

[53] G. Wu, X. Gao, G. Yan and G. Chen, "Parallel greedy algorithm to multiple influence maximization in social network," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 13, pp. 1–21, 2021.

[54] X. Xie, J. Li, Y. Sheng, W. Wang and W. Yang, "Competitive influence maximization considering inactive nodes and community homophily," *Knowledge-Based Systems*, vol. 233, pp. 107497, 2021.

[55] Y. Xue, M. Li, H. Arabnejad, D. Suleimenova, A. Jahani, B. C. Geiger, F. Boesjes, A. Anagnostou, S. J. E. Taylor, X. Liu and D. Groen, "Many-objective simulation optimization for camp location problems in humanitarian logistics," *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 3, art. no. 100017, Sept. 2024.

[56] X. Yi, Z. Wang, S. Liu and Q. Tang, "Acceleration model considering multi-stress coupling effect and reliability modeling method based on nonlinear Wiener process," *Quality and Reliability Engineering International*, vol. 40, no. 6, pp. 3055-3078, May. 2024.

[57] X. Yi, C. Xu, X. Fang and S. Liu, "A new reliability analysis method for software-intensive systems with degradation accumulation effect based on goal oriented methodology," *Quality and Reliability Engineering International*, vol. 40, no. 1, pp. 236-260, May. 2023.

[58] X. Yi and T. Xu, "Distributed event-triggered estimation for dynamic average consensus: A perturbation-injected privacy-preservation scheme," *Information Fusion*, vol. 108, art. no. 102396, Aug. 2024.

[59] X. Yi, H. Yu and T. Xu, "Solving multi-objective weapon-target assignment considering reliability by improved MOEA/D-AM2M," *Neurocomputing*, vol. 563, art. no. 126906, Jan. 2024.

[60] R. Zhang, H. Liu, Y. Liu and H. Tan, "Dynamic event-triggered state estimation for discrete-time delayed switched neural networks with constrained bit rate," *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2334304, 2024.

[61] C. Zhou, P. Zhang, W. Zang and Li Guo, "On the upper bounds of spread for greedy algorithms in social network influence maximization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 10, pp. 2770–2783, 2015.

[62] Y. Zhu, D. Li and Z. Zhang, "Minimum cost seed set for competitive social influence," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, 2016.

[63] Z. Zhao, L. Xia, L. Jiang, Q. Ge and F. Yu, "Distributed bandit online optimisation for energy management in smart grids," *International Journal of Systems Science*, vol. 54, no. 16, pp. 2957–2974, 2023.

Mincan Li received the Ph.D. degree in computer science and technology from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2021. She is currently a postdoctoral researcher in the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. Her research interests include multi-agent systems, data engineering, many-objective optimization and machine learning.

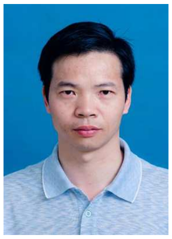
Zidong Wang (SM'03-F'14) received the B.Sc. degree in mathematics from Suzhou University, Suzhou, China, in 1986, and the M.Sc. degree in applied mathematics and the Ph.D. degree in electrical engineering from Nanjing University of Science and Technology, Nanjing, China, in 1990 in 1994, respectively.

He is currently Professor of Dynamical Systems and Computing in the Department of Computer Science, Brunel University London, U.K. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany and the UK. Prof. Wang's research interests include dynamical systems, signal processing, bioinformatics, control theory and applications. He has published a number of papers in international journals. He is a holder of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, William Mong Visiting Research Fellowship of Hong Kong.

Prof. Wang serves (or has served) as the Editor-in-Chief for *International Journal of Systems Science*, the Editor-in-Chief for *Neurocomputing*, the Editor-in-Chief for *Systems Science & Control Engineering*, and an Associate Editor for 12 international journals including *IEEE Transactions on Automatic Control*, *IEEE Transactions on Control Systems Technology*, *IEEE Transactions on Neural Networks*, *IEEE Transactions on Signal Processing*, and *IEEE Transactions on Systems, Man, and Cybernetics-Part C*. He is a Member of the Academia Europaea, a Member of the European Academy of Sciences and Arts, an Academician of the International Academy for Systems and Cybernetic Sciences, a Fellow of the IEEE, a Fellow of the Royal Statistical Society, and a member of program committee for many international conferences.

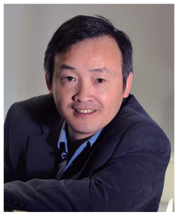
Citation information: DOI: 10.1109/TNNLS.2025.3588236, IEEE Transactions on Neural Networks and Learning Systems

Simon J. E. Taylor received the B.Sc. (Hons.) degree in Industrial Studies and the M.Sc. degree in Computer Studies from Sheffield Hallam University, Sheffield, U.K., in 1986 and 1988, respectively, and the Ph.D. degree in Distributed Simulation from Leeds Metropolitan University, Leeds, U.K., in 1993. He is currently Professor of Computing and Vice-Dean (Research) for the College of Engineering, Design and Physical Sciences at Brunel University. He is also Co-Director of the Modelling & Simulation Group at the same University, a group that has contributed to the area for around 30 years. His research has focused on digital infrastructures and Modelling & Simulation and has impacted over 3 million students and 300 universities in Africa, over 30 SMEs and several large enterprises including the Ford Motor Company and Sellafield. He co-founded the UK Operational Research Society's Journal of Simulation and the UK Simulation Workshop series and continues to be an editor at the Journal. He is a former chair and a member of ACM SIGSIM's Steering Committee, an advisory board member of the NSCU Simulation Archive and executive chair of the international Simulation Exploration Experience. He is has been in various programme committee positions in the IEEE/ACM Winter Simulation Conference for over 20 years and is the General Chair of WSC 2025. He continues to be interested in advances in Modelling & Simulation, international development and Open Science, as well as helping younger colleagues to rapidly start their careers.



Kenli Li received the Ph.D. degree in Computer Science from Huazhong University of Science and Technology, China, in 2003. He was a visiting scholar at the University of Illinois at Urbana-Champaign from 2004 to 2005. He is currently a Cheung Kong Professor of Computer Science and Technology at Hunan University (HNU), the Vice-President of the HNU, the Dean of the College of Information Science and Electronic Engineering of HNU, and the Director in the National Supercomputing Center in Changsha. His major research interests include parallel and distributed processing, high-performance computing, and big data management. He has published more than 350 research papers in international conferences and journals such as IEEE-TC/TPDS, HPCA, ISCA, SC, MM, AAAI, DAC, ICDE, etc. He is a Fellow of the CCF and a Senior Member of the IEEE. He is currently serving or has served as an Associate Editor for IEEE-TC, IEEE-TII, and IEEE-TSUSC.

Xiangke Liao received the B.S. degree in Computer Science and Technology from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 1985, and the M.S. degree in Computer Science and Technology from the National University of Defense Technology, Changsha, China, in 1988. He is currently a Full Professor and the Dean of the College of Computer Science, National University of Defense Technology. His research interests include parallel and distributed computing, high-performance computer systems, operating systems, cloud computing, and networked embedded systems.



Xiaohui Liu received the B.Eng. Degree in Computing from Hohai University, Nanjing, China, in 1982 and the Ph.D. degree in Computer Science from Heriot-Watt University, Edinburgh, U.K, in 1988. He is currently Professor of Computing at Brunel University London where he conducts research in Artificial Intelligence, Data Science and Optimization, with applications in diverse areas including biomedicine and engineering.