World Journal of Information Technology
Print ISSN: 2959-9903
Online ISSN: 2959-9911
DOI: https://doi.org/10.61784/wjit3089

A COLLABORATIVE DECISION-MAKING FRAMEWORK FOR INFLUENCE MAXIMIZATION BASED ON GRAPH CONTRASTIVE LEARNING AND DEEP REINFORCEMENT LEARNING

JiaWei Kang
School of Artificial Intelligence, Shenyang Normal University, Shenyang 110000, Liaoning, China.

Abstract: To address the limitations of traditional influence maximization methods in complex social networks, including insufficient feature representation, strong dependence on predefined diffusion models, and limited capability for collaborative seed selection, this paper proposes a collaborative decision-making framework based on graph contrastive learning and deep reinforcement learning. First, the social network is modeled as a directed weighted graph, and a mathematical formulation of influence propagation and seed selection is established under the Independent Cascade(IC)model. Then, a graph contrastive learning strategy is introduced to conduct self-supervised pre-training over network topology, edge weights, and node attributes. By leveraging multi-view graph augmentation and node-level contrastive objectives, the framework learns high-quality node representations that better characterize diffusion potential. On this basis, the influence maximization task is reformulated as a sequential decision-making process, and a Double DQN-based seed selection mechanism is designed to iteratively select seed nodes. Marginal influence gain is adopted as the immediate reward to enable dynamic collaborative optimization of the seed set. Experiments conducted on six real-world social network datasets, including Petster-hamster, Tv-show, Politician, Advogato, Public, and Epinions, demonstrate that the proposed method consistently outperforms Random, PageRank, gIM, S2V-DQN, and ToupleGDD under different seed budgets. The results verify that combining the representation advantages of graph contrastive learning with the sequential decision-making capability of deep reinforcement learning can effectively improve influence maximization performance in complex networks.
Keywords: Influence maximization; Graph contrastive learning; Deep reinforcement learning

1 INTRODUCTION

Influence Maximization(IM), a pivotal challenge in social network analysis, information dissemination, and control systems, aims to identify an optimal set of nodes as initial propagation sources within a given budget(typically the number of seed nodes k) [1,2]. The goal is to ensure that information, opinions, or behaviors can reach the widest possible audience through a specific diffusion mechanism. This problem has extensive practical applications in viral marketing, public opinion guidance, public health interventions, and innovation promotion.

Traditional methods for maximizing influence primarily focus on greedy algorithms optimized for submodularity, heuristic centrality metrics, and Monte Carlo simulations based on fixed diffusion models(e. g. , Independent Cascade Model[IC] and Linear Threshold[LT]Model) [1,3,4]. These approaches have yielded substantial theoretical achievements and practical validation in ideal scenarios featuring relatively static structures, homogeneous node and edge attributes, pre-defined diffusion mechanisms, and known parameters [5]. However, real-world networks often present challenges such as high-dimensional complexity, heterogeneous node and edge attributes, uncertain or dynamically changing diffusion mechanisms, incomplete observational data, evolving network topologies over time, and prohibitively high computational costs due to massive network scales [5]. In such scenarios, traditional methods frequently encounter limitations including insufficient scalability, weak generalization capabilities, and excessive dependence on diffusion models and parameter assumptions, which restrict their applicability and effectiveness in practical complex systems.

In recent years, the rapid development of deep learning, particularly Graph Neural Networks(GNNs), has provided a novel approach to directly learning complex mappings between network structures, node attributes, and propagation behaviors from data [6]. Compared to traditional methods relying on explicit diffusion models and manual parameter tuning, GNNs enable end-to-end representation learning, automatically capturing latent associations between node influence and local-global topological structures [7-9]. This offers a data-driven modeling framework for influence maximization problems where diffusion mechanisms are uncertain or environment-dependent. The advantages of GNNs are twofold:First, they demonstrate strong cross-network transferability, allowing rapid adaptation and model reuse across networks of varying scales and domains, thereby reducing dependence on manual feature engineering for single networks [9,10]. Second, while traditional greedy algorithms in large-scale networks typically require extensive propagation simulations or marginal gain estimation with significant computational overhead, GNN-based node influence prediction can achieve results through a single or minimal forward inference, substantially reducing time-consuming repeated simulations and enabling near-real-time decision-making.

This study addresses the limitations of traditional influence maximization methods in complex real-world networks by exploring a graph neural network-based approach for influence prediction and seed node selection [11]. We propose a novel framework that integrates GNN representation learning with influence maximization tasks, enabling efficient, scalable, and generalizable solutions without relying on strong diffusion model assumptions. Our research not only advances the theoretical development of influence maximization in open environments but also provides a new technical pathway for designing rapid and adaptive information dissemination strategies in practical applications.

2 PROBLEM MODELING

2.1 Background Knowledge

Abstraction of Network as a Directed Weighted Graph

$$G = (V, E, W), \quad |V| = N, \ |E| = M \tag{1}$$

where $V=\{v_1,\ldots,v_N\}E\subseteq V\times VW=\{w_{uv}\}_{(u,v)\in E}$is the set of nodes and is the set of directed edges;is the edge weight, which characterizes the propagation strength of node u to node v(e. g. , interaction frequency, link quality, or trustworthiness). Let be the set of directed neighbors.

$$\mathcal{N}^{+}(u) = \{v|(u,v) \in E\}, \quad \mathcal{N}^{-}(v) = \{u|(u,v) \in E\} \tag{2}$$

If node attributes/content features exist, they $X\in\mathbb{R}^{N\times d}$are represented by a matrix;if no explicit features are available, X can be constructed using structural statistics(e. g. , degree, centrality, position coding, etc.). For the convenience of reuse in subsequent chapters, the core symbols used in this paper are summarized in Table 1.

The maximization of influence depends on information diffusion dynamics. In this study, the independent cascade(IC) model is adopted as the core diffusion mechanism, abstracting propagation as a discrete-time stochastic process. By parameterizing the propagation probability with learnable models tailored to real-world directed weighted networks, we establish a realistic simulation environment for subsequent reinforcement learning-based seed selection.

1) IC diffusion mechanism. Given a seed set S, the initial activation set isA_0=S.

At discrete time t=0, 1, 2, . . . , if node u is first activated at time t, it initiates a propagation attempt to each $v\in N^{+}(u)\backslash A_t$unactivated outgoing neighbor at the next time, with probability puv to activate node v. This attempt occurs only once, and regardless of success, u will not attempt to activate v again thereafter. Thus, the conditional probability that node v remains unactivated at time t+1 can be expressed as

$$\mathbb{P}(v \notin A_{t+1}|A_t) = \prod_{u\in\mathcal{P}_t(v)} (1 - p_{uv}) \tag{3}$$

All nodes successfully activated during this time step will join the activation set and become new potential transmitters in the next time step t+1. The probability of node v being activated at t+1 depends on whether at least one node in Pt(v)—the set of all initially activated and connected nodes at time t—has successfully transmitted. Here, Pt(v)denotes the set of nodes initially activated at time t that satisfy$(u, v)\in$E(i. e. , the"newly activated parent nodes"that will attempt to activate v in this step). This yields

$$\mathbb{P}(v \in A_{t+1}|A_t) = 1 - \prod_{u\in\mathcal{P}_t(v)} (1 - p_{uv}) \tag{4}$$

This model visually demonstrates how independent attempts from multiple parent nodes generate cumulative effects. The diffusion process continues until a specific time T when no new$|A_T(S)|$nodes are activated, marking the process's termination. The resulting activation set, whose size(i. e. , the coverage of influence)serves as the core metric for evaluating the quality of the seed set S.

2) Weight mapping of transmission probability. A key contribution of this study lies in the parametric modeling of transmission probability puv. In real-world networks, the interaction strength or association degree between nodes can often be observed as edge weights wuv, such as communication frequency, $\phi:\mathbb{R}\to(0,1)$trust weight, or link quality. However, the actual transmission probability is difficult to obtain directly. To address this, we introduce a monotonic mapping function defined by parametersαandβ, which maps edge weights to transmission probability:

$$p_{uv} = \phi(w_{uv}) = \sigma(\alpha w_{uv} + \beta), \quad \sigma(z) = \frac{1}{1+e^{-z}} \tag{5}$$

Here, αandβare adjustable parameters(determined through historical diffusion logs, validation sets, or empirical settings). When edge weights are normalized to[0, 1], we may alternatively set $p_{uv}=w_{uv}$ as a simplified approach.

The core of the influence maximization problem lies in quantifying the propagation effect of the selected seed set andestablishing a solvable optimization model based on this. Given a seed set in an independent cascade model, information propagation will cease after several time steps, with the activated node set at this point denoted as$A_T(S)$. We define the influence function$\sigma(S)$ as the expected value of the activated node scale, i. e. , $\sigma(S)=E[|A_T(S)|]$,
The expectation averages the randomness in the diffusion process(whether each edge spreads successfully). This function intuitively reflects the average number of nodes covered by the seed set S under random diffusion.
Based on the above definition, the classic influence maximization problem can be formulated as a constrained combinatorial optimization $S\subseteq V$ problem:given a seed budget K, the objective is to find a subset of nodes that maximizes the expected influence.

$$\max_{S\subseteq V} \sigma(S) \quad s.t. \quad |S| \le K \tag{6}$$

This problem NP−is classified as a difficult one, and the function$\sigma(S)$is generally intractable to compute analytically, primarily due to the stochastic nature of the diffusion process and the often complex network architecture. To evaluate$\sigma(S)$in practice, this study employs Monte Carlo simulation for estimation. Specifically, we independently run R diffusion simulations, where the set of activated nodes at the end of the r-th simulation is denoted as$A_T^{(r)}(S)$, and the influence estimate is defined as

$$\hat{\sigma}(S) = \frac{1}{R}\sum_{r=1}^{R} \left|A_T^{(r)}(S)\right| \tag{7}$$

This estimator serves as a unbiased estimate of$\sigma(S)$and converges as R increases. In the subsequent sequential decision-making and reinforcement learning framework, we will utilize this estimate to approximate the instant reward at each step, thereby driving policy learning.

Table 1 Main Symbol Explanation

symbol	meaning
G=(V, E, W)	directed weighted network
N, M	Number of nodes and edg
w_{uv}	The weight of the edge(u, v)
p_{uv}	The success probability of propagation for the edge(u, v)
S	Seed set(the set of nodes initially activated)
K	Seed Budget($|S|\le K$)
A_t	The set of active nodes at time t
$\sigma(S)$	The expected influence(expected coverage)of the seed set S
$\hat{\sigma}(S)$	The Influence of Monte Carlo Estimation

2.2 Decision-making Perspective of Serialized Site Selection

To facilitate integration with subsequent deep reinforcement learning(Deep Q-Network, DQN)algorithm frameworks, the combinatorial optimization problem of maximizing influence is re-formulated as a sequential decision-making process. This transformation aims to decompose the overall task of'selecting K seeds from the node set V'into K sequential decision steps. Specifically, the decision step t=1,2,…,Kcount is defined. At step t, the agent selects a node at from the unselected node set, i. e. :

$$a_t \in V \setminus S_{t-1} \tag{8}$$

Here, St denotes the cumulative set of seeds selected over the first t steps, satisfying the recursive relationship:

$$S_t = S_{t-1} \cup \{a_t\}, \quad S_0 = \emptyset, \quad |S_K| = K \tag{9}$$

For each decision step, the marginal influence gain$\Delta_t$of the current seed set St relative to the previous setS_{t-1}can be calculated, defined as:

$$\Delta_t \triangleq \sigma(S_t) - \sigma(S_{t-1}) \tag{10}$$

It corresponds to"the additional coverage generated by the current action. "In practice, Monte Carlo estimation can be used to replace $\sigma(\cdot)$. This serialized modeling enables the subsequent construction of a reinforcement learning framework using state representations(graph structure and selected sets), action space(unselected nodes), and rewards(marginal gains), thereby achieving seamless integration with DQN.

3 A COLLABORATIVE DECISION-MAKING FRAMEWORK BASED ON GRAPH CONTRASTIVE LEARNING AND DEEP REINFORCEMENT LEARNING

3. 1 Contrastive Learning vs. Representational Learning

3.1.1 General approach

In the problem of influence maximization, identifying key nodes with high propagation potential is the core challenge. Directly learning node and graph structure state representations in an end-to-end manner during the reinforcement learning phase often leads to issues such as training instability, low sample efficiency, and insufficient generalization ability. To address this, this study proposes introducing Graph Contrastive Learning(GCL)as a self-supervised pre-training method before reinforcement learning. This approach aims to extract high-quality, transferable node representations from network topology, edge weight information, and node attributes, which serve as state input features for subsequent DQN decision-making.

Specifically, given a directed weighted graph G=(V, E, W)and its node $X\in\mathbb{R}^{N\times d}$ feature matrix(constructed from structural statistics like node degree or centrality if no explicit features are provided), the goal of contrastive learning is to train an encoder without labeled data:

$$f_\theta: (G, X) \mapsto H = [h_1, ..., h_N]^\top \in \mathbb{R}^{N\times d_h} \tag{11}$$

This ensures that nodes with semantic or structural similarity are clustered together in the representation space, while dissimilar nodes are separated. The proposed method adheres to the standard paradigm of"dual-view augmentation+node-level contrastive loss".

3.1.2 Graph augmentation and multi-view construction

The core of graph contrastive learning is to generate multiple semantically consistent yet morphologically distinct views from a single input graph, thereby achieving self-supervised learning by maintaining similarity in representations of the same node across $k\in\{1,2\}$ different views. Specifically, we independently construct two randomly augmented views for each original graph, denoted as the i-th view:

$$\tilde{G}^{(k)} = \left(V, \tilde{E}^{(k)}, \tilde{W}^{(k)}\right), \quad \tilde{X}^{(k)} = \mathcal{T}_X^{(k)}(X) \tag{12}$$

In this $T_E^{(k)}$, $T_W^{(k)}$and$T_X^{(k)}$ paper, we design three kinds of reproducible augmentation strategies to construct positive samples with different characteristics, which can maintain the macro statistics and the distribution of propagation intensity of the network, and introduce moderate randomness.

First, random edge dropout is employed to simulate network link uncertainty:each edge is $(u,v)\in E$ removed with a fixed probability$\rho_E$to obtain a sparse view, i. e.

$$\mathbb{P}((u, v) \in \tilde{E}^{(k)}) = 1 - \rho_E \tag{13}$$

Secondly, a weight perturbation is applied to the retained edges, using multiplicative Gaussian noise to simulate fluctuations in propagation probability estimates, with non-negative truncation applied to the results.

$$w_{uv}^{(k)} = max\{0, w_{uv} \cdot (1 + \epsilon_{uv}^{(k)})\}, \quad \epsilon_{uv}^{(k)} \sim \mathcal{N}(0, \sigma_w^2) \tag{14}$$

Furthermore, random feature masking is implemented at the node level, where the feature matrix is randomly zeroed across all dimensions.

$$\tilde{X}^{(k)} = X \odot M^{(k)}, \quad M_{ij}^{(k)} \sim Bernoulli(1 - \rho_X) \tag{15}$$

It represents$\odot$element-by-element multiplication.

The three augmentation operations mentioned above can be used independently or in combination to form the foundation for multi-view generation. This method constructs'different observations of the same semantics'through controllable random transformations while preserving the semantic integrity of nodes and edges, thereby effectively supporting the positive sample alignment objective in subsequent node-level contrastive learning.

3.1.3 Image encoder and projection head

In the graph contrastive learning framework, the encoder maps the enhanced graph structure and its node features into low-dimensional, dense vector representations. For each, For the augmented $(\widetilde{G}^{(k)},\widetilde{X}^{(k)})$ view, we obtain the node embedding matrix using the shared-parameter graph encoderf_θ.

$$H^{(k)} = f_\theta \setminus big(\widetilde{G}^{(k)}, \widetilde{X}^{(k)} \setminus big) \in \mathbb{R}^{N\times d_h} \tag{16}$$

Here, d_h denotes the embedding dimension. To effectively encode both local and global node information, this study employs a message-passing-based graph neural network as the encoder, explicitly modeling the influence of directed weighted edges through a weighted aggregation mechanism.

Specifically, we employ a universal Graph Neural Network(GNN)architecture that supports edge-weight input. Taking the l-th layer as an example, the representation of node$v_i$is updated as follows:

$$h_i^{(l)} = \sigma\left(W^{(l)}h_i^{(l-1)} + \sum_{j\in\mathcal{N}^-(i)} \alpha_{ji}^{(l)}\, U^{(l)}h_j^{(l-1)}\right), \quad h_i^{(0)} = x_i \tag{17}$$

Here, $\sigma(\cdot)$denotes a nonlinear activation function, with$W^{(l)}$and$U^{(l)} \in \mathbb{R}^{d_h\times d_h}$being trainable parameters. $N^-(i)$represents the set of incoming neighbors for node vi. The weight coefficient$\alpha_{ii}^{(l)}$modulates message contributions from different neighbors, with its design fully incorporating edge weight information. For instance, it can be defined in a normalized form:

$$\alpha_{ji}^{(l)} = \frac{\widetilde{w}_{ji}}{\sum_{j'\in\mathcal{N}^-(i)} \widetilde{w}_{j'i} + \epsilon} \tag{18}$$

In contrast, contrastive learning typically incorporates a projection head after embedding to enhance the separability of the representation space. Let the projection head be denoted asg_ϕ, then

$$Z^{(k)} = g_\phi(H^{(k)}) \in \mathbb{R}^{N\times d_z}, \quad z_i^{(k)} = \frac{Z_i^{(k)}}{\|Z_i^{(k)}\|_2} \tag{19}$$

It is the$z_i^{(k)}$unit norm vector, which is convenient for using cosine similarity.

3.1.4 Node-level comparison target(InfoNCE)

The core objective of node-level contrastive learning is to develop a representation that brings nodes with similar features closer together in different augmented views, while keeping distinct nodes farther apart in the representation space. This study constructs the node-level contrastive learning objective by using"representations of the same node across different views"as positive pairs and"different nodes"as negative pairs. The similarity function employs cosine similarity.

$$sin(z_a, z_b) = z_a^\top z_b \tag{20}$$

For any node, v_i1→2define the InfoNCE loss from the view as

$$\ell_i^{(1\to2)} = -\,log\frac{exp\left(sin(z_i^{(1)}, z_i^{(2)})/\tau\right)}{\sum_{j=1}^{N} exp\left(sin(z_i^{(1)}, z_j^{(2)})/\tau\right)} \tag{21}$$

Here, $\tau>0$ denotes the temperature coefficient. This parameter regulates the model's focus on challenging negative samples:a smallerτenhances the model's ability to distinguish similar samples, whereas a largerτmakes the loss function more smoothing toward similarity differences.

Similarly, the loss from View 2 to View 1 can be symmetrically$l_i^{(2\to1)}$defined. The final node-level contrast loss is the average of these two directional losses.

$$\mathcal{L}_{CL} = \frac{1}{2N}\sum_{i=1}^{N}\left(\ell_i^{(1\to2)} + \ell_i^{(2\to1)}\right) \tag{22}$$

This symmetric design ensures fairness in the learning process for both augmented views, thereby enhancing$\mathcal{L}_{CL}$training stability. By minimizing, the model is incentivized to map multi-view representations of the same node to adjacent positions in the embedded space while pushing representations of different nodes apart.

3.1.5 Pre-training output and downstream interfaces

After completing pre-training, this paper retains $H=f_\theta(G,X)g_\phi h_i\{h_i\}_{i=1}^N$ the encoder output as node representations(the projection head$g_\phi$can be discarded in downstream stages)and uses$h_i$as the foundational representation for nodev_i. The objective of this phase is to properly preserve and format the pre-training results, providing stable and information-rich state inputs for the subsequent reinforcement learning decision module. In the following reinforcement learning sections, state inputs will be constructed based on $\{h_i\}_{i=1}^N$ representation(e. g, concatenating with"selected set indicator vector/remaining budget")to implement a joint framework of"contrastive learning representation+DQN decision".

3.2 Reinforcement Learning Algorithm: Sequential Seed Selection Based on DQN

3.2.1 MDP definition: state, action, transition, and reward

To address the problem of maximizing influence within the reinforcement learning framework, this paper constructs a corresponding Markov Decision $M=\langle S,A,P,r,\gamma\rangle$ Process(MDP). We define, transforming the portfolio optimization task into a sequence decision process of length K. The following sections elaborate on the five key components of the MDP.

State:The state at step t requires $s_t \in Sm_t \in \{0,1\}^N$ simultaneous encoding of the currently selected node, network structure, and remaining decision information. This paper utilizes the node representation matrix $H=[h_1,\ldots,h_N]^T$ pre-trained in Chapter 2 as static graph features, and introduces a selection indicator vector$m_t \in \{0,1\}^N$whose i-th dimension is defined as

$$\left((m_t)_i = \begin{cases} 1, & v_i \in S_t, \\ 0, & v_i \notin S_t, \end{cases}\right) \quad S_t = \{a_1, \ldots, a_t\} \tag{23}$$

The vector mt explicitly records the nodes selected as seeds up to time t. Additionally$b_t=K-t$, let denote the remaining number of selectable nodes(residual budget). Thus, the complete state can be represented as a triplet.

$$s_t \triangleq (H, m_t, b_t), \quad b_t = K - t \tag{24}$$

This representation integrates the network's structural semantics(H), decision history(m_t), and resource constraints(b_t), providing sufficient information for the subsequent Q network to evaluate the potential value of each candidate action.

Action:At each step t, the agent selects a new seed from the unselected node set, meaning the action space is

$$\mathcal{A}(s_t) = V \setminus S_t, \quad a_t \in \mathcal{A}(s_t) \tag{25}$$

To prevent redundant selections, both training and inference phases employ action masking, where the Q values of selected nodes are set to$-\infty$to ensure action validity.

After the transition executes the action at, the seed set is updated to

$$S_{t+1} = S_t \cup \{a_t\}, \quad m_{t+1} = m_t + e_{a_t} \tag{26}$$

Here, $e_{a_t}$is a one-hot vector with a corresponding dimension of 1 in the at dimension. Since H is fixed within an episode, the stochasticity of MDP mainly originates from the sampling of the diffusion process during reward estimation.

To align the reward with the original target$\sigma(S)$, this study employs marginal impact gain as the immediate reward.

$$r_t \triangleq \sigma(S_{t+1}) - \sigma(S_t) \tag{27}$$

Given that$\sigma(\cdot)$is non-analytic, Monte Carlo estimation$\hat{\sigma}(\cdot)$is used in practice to obtain a computable reward.

$$\hat{r}_t \triangleq \hat{\sigma}(S_{t+1}) - \hat{\sigma}(S_t) \tag{28}$$

This definition ensures that cumulative returns align with ultimate impact:

$$\sum_{t=0}^{K-1} r_t = \sigma(S_K) - \sigma(S_0) = \sigma(S_K) \tag{29}$$

Discount Factor(Discount)and Termination Condition:Given the fixed decision$\gamma \in (0,1]$process length K, the discount factor can be set. To directly maximize the final coverage scale, is typically set to 1. The episode terminates when the selected seed count reaches K(i. e. , t=K)or when no non-empty actions are available.

3.2.2 Q network structure and action scoring

In the DQN-based sequential decision framework, the Q network's core function is to accurately evaluate the long-term expected return of selecting each candidate action under given states. To model the discrete action space of node selection, this paper designs a node-scoring Q network. The network treats each candidate node as an independent action and outputs a corresponding Q-value score, reflecting the expected cumulative influence gain of choosing that node as a seed in the current state.

Specifically, let the parameter of network Q beψ. For state st and candidate node vi(i. e. , action at=i), its Q value is calculated through a differentiable scoring functionq_ψ:

$$Q_\psi(s_t, a_t = i) = q_\psi(h_i, u_t) \tag{30}$$

where$u_t$denotes the global context feature, which can be aggregated from the selected set, for example

$$\bar{h}_t = \frac{1}{|S_t|+\epsilon}\sum_{v_j \in S_t} h_j, \quad u_t = [\bar{h}_t; b_t] \tag{31}$$

Here denotes$[\cdot\ ;\cdot\ ]$the vector concatenation operation, where$\epsilon$is a small constant introduced for numerical stability. This design enables the Q network to not only evaluate the value of individual nodes but also perceive the overall composition of selected nodes and the decision-making process(remaining steps), thereby avoiding the selection of redundant or low-collaborative-effect nodes.

Action Masking:To prevent the policy from repeatedly selecting the same node, an action masking mechanism must be implemented$mask_t \in \{0,1\}^N$. Define a masking vector, where the i-th element is:

$$mask_t(i) = 1 - (m_t)_i \tag{32}$$

At each decision step, only nodes satisfying$mask_t(i)=1$ (i. e. , unselected)are considered valid action candidates. In practice, this is typically achieved by assigning an extremely$-\infty$large negative value(e. g. , -1)to invalid actions(selected nodes)or setting their Q values to zero and excluding them from the maximum value search to implement filtering.

3.2.3 DQN learning objectives and parameter updates

To achieve stable and efficient training of deep Q networks, this paper employs a replay buffer and target network for stable training. Let the online network parameters beψ, the target network parameters beψ^-, and the replay pool be D. Each interaction generates a transfer sample.

$$(s_t, a_t, \hat{r}_t, s_{t+1}) \in D \tag{33}$$

TD target for DQN, where TD target is defined as

$$y_t = \hat{r}_t + \gamma max_{a' \in \mathcal{A}(s_{t+1})} Q_{\psi^-}(s_{t+1}, a') \tag{34}$$

Then minimize the mean squared TD error:

$$\mathcal{L}_{DQN}(\psi) = \mathbb{E}_{(s,a,\hat{r},s')\sim\mathcal{D}}[(Q_\psi(s, a) - y)^2] \tag{35}$$

To address the overestimation of Q-values caused by the maximization operation in standard DQN, this paper proposes the Double DQN mechanism. The core idea is to decouple action selection from value evaluation, where the online network selects actions while the target network assesses their value.

$$a^* = arg \max_{a' \in \mathcal{A}(s_{t+1})} Q_\psi(s_{t+1}, a'), \quad y_t^{DDQN} = \hat{r}_t + \gamma Q_{\psi^-}(s_{t+1}, a^*) \tag{36}$$

Replace yt withy_t^{DDQN}in the formula to obtain more accurate and stable value estimates.

Target network update. The target network uses soft or hard updates. The hard update method synchronizes every C steps:

$$\psi^- \leftarrow \psi \quad every C steps.$$

3.2.4 Exploration strategies and action masking

During the training phase, theϵ-greedy strategy is employed for exploration. At states_t, the agent uniformly samples from the set of available actions with probabilityϵ, otherwise selects the action with the highest Q-value.

$$a_t = \begin{cases} Uniform(\mathcal{A}(s_t)), & w.p.\ \epsilon, \\ arg\ \max_{a \in \mathcal{A}(s_t)} Q_\psi(s_t, a), & w.p.\ 1-\epsilon. \end{cases} \quad (37)$$

Here, $A(s_t)=V\backslash S_t$ implicitly incorporates action masking;during implementation, invalid actions can be assigned the value$-\infty$to prevent selection.

3.2.5 Training process and inference strategy

The training process begins with an empty set: $S_0=\varnothing$. For t=0, . . . , K−1:(1)Select at from the unselected nodes according toϵ-greedy;(2)Update the set to obtain S_{t+1} ;(3)Estimate the reward $\hat{r}_t$ via Monte Carlo;(4)Store in the replay pool and perform gradient update onψ;(5)Periodically update the target network.

In the inference strategyϵ=0, $S_0 = \emptyset$testing phase, greedily select sequentially from the initial set.

$$a_t = arg\ \max_{a \in \mathcal{A}(s_t)} Q_\psi(s_t, a), \quad S_{t+1} = S_t \cup \{a_t\} \quad (38)$$

The final output is$S_K$as the seed set.

4 EXPERIMENT AND ANALYSIS

4.1 Experiment Setup

4.1.1 Experimental environment

We verify that the edge weight settings remain consistent across both validation and test datasets. To simulate networks with incomplete observations, we processed all datasets by randomly discarding 50%of edges and 50%of node features, thereby modeling missing connections and partially observed/noisy attribute information. All models were evaluated under these conditions. For each test dataset, we varied the budget parameter b to the values{10, 20, 30, 40, 50}. All experiments were conducted on the following hardware configuration:Intel Xeon Platinum 8350H CPU(2. 60 GHz, 48 cores), 384GB DDR4 memory, NVIDIA RTX 4090 GPU(24 GB video memory), and Ubuntu 20. 04 operating system.

4.1.2 Experimental data set

The experiment utilized six real-world social network datasets for validation:Petster-hamster, Tv-show, Politician, Advogato, Public, and Epinions. These datasets encompass social networks of varying scales and topologies, enabling comprehensive evaluation of the algorithm's effectiveness across diverse scenarios. All datasets were abstracted as directed weighted graphs, where nodes represent social network users, edges represent user interactions, and edge weights characterize the transmission intensity between nodes.

4.1.3 Comparison algorithm

To evaluate the performance of our collaborative decision-making algorithm(ours)based on graph contrastive learning and deep reinforcement learning, we selected five representative influence-maximizing algorithms for comparison, including random selection, classical graph algorithms, traditional greedy algorithms, deep learning algorithms, and advanced heuristic algorithms, as detailed below:

1)Random:Select k nodes randomly as the seed set to serve as the baseline algorithm.

2)PageRank:The system calculates node importance using the PageRank algorithm and selects the top k ranked nodes as the seed set.

3)gIM:A classical greedy algorithm for influence maximization, which selects seed nodes based on marginal gain.

4)S2V-DQN:A deep learning algorithm that combines graph-structured vectors with deep Q networks, where graph encoding is integrated with reinforcement learning for seed selection.

5)ToupleGDD:An advanced heuristic algorithm based on graph depth discounting, which fully exploits the propagation characteristics of network topology for seed selection.

4.1.4 Experimental parameters and evaluation indicators

In the experiment, the seed set size k was set to 10, 20, 30, 40, and 50 to cover different seed budget scenarios. All algorithm experimental results were averaged to avoid random errors. As shown in Tables 2, 3, and 4.

The core evaluation metric for algorithm performance is the expected influence range of the seed set, which represents the anticipated number of nodes activated by the seed set under the Independent Cascade(IC)propagation model. This metric directly reflects the propagation capability of the seed set, with higher values indicating superior algorithm performance.

In the proposed algorithm, the edge dropout probability for graph contrastive learning is set to 0. 2, the Gaussian noise variance for edge weight perturbation is 0. 1, and the feature mask probability is 0. 2. The reinforcement learning employs the Double DQN framework with an experience replay pool size of 10, 000, where the target network undergoes hard updates every 50 steps. The exploration probability of theε-greedy policy decays linearly from 0. 9 to 0. 1. The propagation model uses the IC model, with propagation probabilities mapped from edge weights via sigmoid functions. Monte Carlo simulations are conducted 100 times to ensure accurate influence estimation.

Table 2 Comparison of Performance across Different Methods in the Petter-hamster and Tv-show Tasks

Method	10_Petster-hamster	20_Petster-hamster	30_Petster-hamster	40_Petster-hamster	50_Petster-hamster	10_Tv-show	20_Tv-show	30_Tv-show	40_Tv-show	50_Tv-show
Random	35. 882	38. 091	92. 167	130. 848	94. 199	23. 779	64. 548	91. 484	79. 532	172. 343
PageRank	12. 896	45. 164	64. 650	75. 665	94. 225	17. 405	27. 414	43. 418	53. 778	63. 649
gIM	151. 627	220. 516	249. 941	261. 548	274. 339	82. 282	166. 036	228. 168	264. 971	361. 307
S2V-DQN	168. 028	177. 726	231. 678	232. 784	274. 242	164. 211	190. 851	234. 212	274. 340	300. 786
ToupleGDD	417. 573	442. 583	423. 213	483. 310	495. 088	533. 662	810. 806	987. 575	1081. 138	1141. 417
ours	429. 128	467. 770	485. 284	501. 256	542. 772	831. 042	1185. 241	1268. 607	1453. 595	1511. 800

Table 3 Comparison of Performance across Different Methods in the Politician and Advogato Tasks

Method	10_Politician	20_Politician	30_Politician	40_Politician	50_Politician	10_Advogato	20_Advogato	30_Advogato	40_Advogato	50_Advogato
Random	52. 571	41. 217	74. 664	175. 115	124. 474	28. 820	116. 038	216. 086	102. 605	181. 823
PageRank	17. 694	28. 408	90. 691	105. 476	120. 287	95. 683	166. 445	286. 553	372. 586	434. 395
gIM	508. 626	650. 391	923. 691	1128. 736	1205. 080	2844. 721	2857. 174	2864. 499	2872. 658	2873. 647
S2V-DQN	160. 208	756. 044	1044. 439	1082. 533	1303. 446	3672. 818	3667. 824	3665. 064	3667. 710	3665. 346
ToupleGDD	2580. 649	2762. 888	2974. 889	3012. 336	3034. 699	3668. 109	3669. 642	3676. 214	3676. 522	3681. 131
ours	2989. 080	3103. 340	3180. 029	3245. 112	3326. 071	3703. 583	3696. 524	3712. 415	3712. 992	3699. 495

Table 4 Comparison of Performance across Different Methods in the Public and Epinions Tasks

Method	10_Public	20_Public	30_Public	40_Public	50_Public	10_Epinions	20_Epinions	30_Epinions	40_Epinions	50_Epinions
Random	17. 314	48. 715	111. 968	118. 112	251. 815	20. 796	58. 344	134. 245	141. 756	302. 062
PageRank	25. 984	58. 487	68. 834	93. 559	106. 696	31. 176	70. 222	82. 638	112. 129	227. 938
gIM	904. 011	1068. 999	2040. 044	2247. 288	2293. 566	597. 833	701. 105	1350. 130	1492. 253	1511. 195
S2V-DQN	785. 534	978. 052	1536. 236	1529. 346	1577. 484	1180. 254	1180. 212	1202. 035	1604. 828	2156. 532
ToupleGDD	5395. 573	5438. 112	5499. 155	5579. 650	5699. 642	3525. 181	3587. 509	3657. 816	3662. 657	3693. 176
ours	5510. 443	5766. 469	5815. 259	5930. 529	6037. 037	3640. 088	3777. 726	3853. 472	3949. 320	3976. 297

4.2 Experimental Results and Performance Analysis

The experiment evaluated the expected impact ranges of our algorithm and five benchmark algorithms across six real-world social network datasets, with varying seed set sizes(k). The results (summarized in the table) demonstrate consistent performance patterns. The analysis is structured around three dimensions:overall trends, dataset-specific performance, and algorithmic comparisons. As shown in Figure 1.

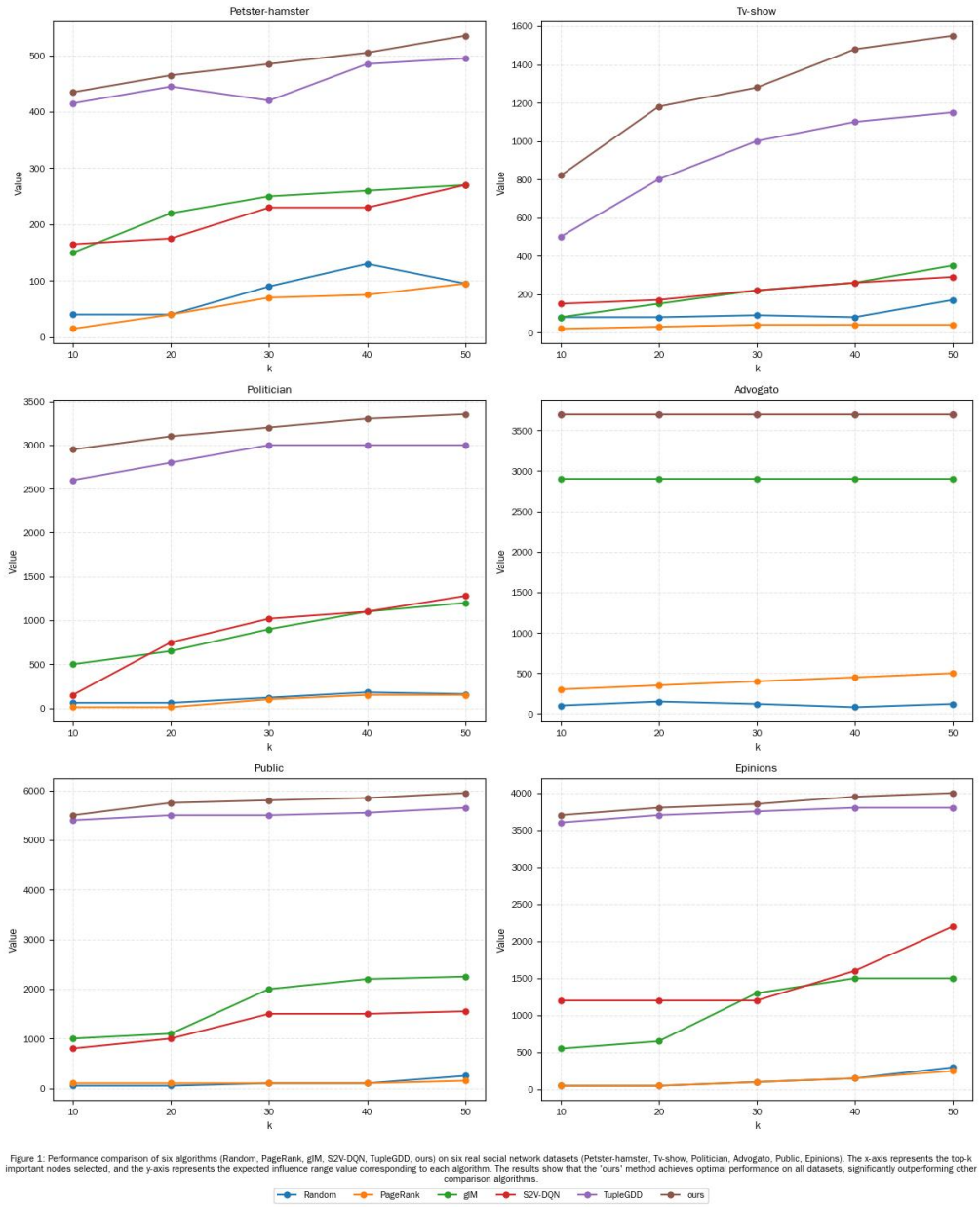

Figure 1: Performance comparison of six algorithms (Random, PageRank, gIM, S2V-DQN, TupleGDD, ours) on six real social network datasets (Petster-hamster, Tv-show, Politician, Advogato, Public, Epinions). The x-axis represents the top-k important nodes selected, and the y-axis represents the expected influence range value corresponding to each algorithm. The results show that the 'ours' method achieves optimal performance on all datasets, significantly outperforming other comparison algorithms.

Figure 1 Comparison of Different Methods on Various Datasets

4.2.1 Overall performance trends

Across all datasets and comparison algorithms, the expected influence range of seed sets increases with the growth of seed set size k, aligning with the fundamental principle of influence maximization:more seed nodes create a broader information dissemination base, thereby activating more network nodes. However, for some algorithms, the growth rate of influence range slows down after k reaches a certain threshold. This occurs because node propagation in networks exhibits overlap, where the marginal influence gain from adding new seed nodes diminishes as the number of seeds increases. This phenomenon becomes more pronounced in datasets with denser topological structures, such as Advogato and Public.

4.2.2 Algorithm performance across different data sets

Despite the relatively small scale of nodes and edges in small datasets(Petster-hamster and Tv-show)and the relatively simple network topology, our algorithm still demonstrates significant advantages. In the Petster-hamster dataset, when k=50, our algorithm achieves an influence range of 542. 772, representing a 9. 6%improvement over the suboptimal ToupleGDD algorithm. In the Tv-show dataset, with k=50, our algorithm achieves an influence range of 1511. 8, showing a 32. 4%improvement over ToupleGDD, marking the largest enhancement among all datasets. This is attributed to the significant variation in edge weight distribution in the Tv-show dataset. Our algorithm's graph contrastive learning module effectively extracts propagation characteristics of edge weights, while the reinforcement learning module enables collaborative selection of seed nodes, significantly enhancing propagation efficiency.

The medium-scale datasets(Politician and Epinions)exhibit moderate topological complexity with diverse propagation paths between nodes. In the Politician dataset, the algorithm achieves an influence range of 3, 326. 071 at k=50, representing a 9. 6%improvement over the ToupleGDD algorithm. For Epinions, the influence range reaches 3, 976. 297 at k=50, showing a 7. 7%enhancement. The pre-trained node representation in this algorithm accurately captures nodes'propagation potential, while the serialized seed selection strategy effectively avoids redundant node selection, demonstrating stable performance advantages in medium-complexity networks.

Large-scale datasets(Advogato and Public)feature extensive nodes and edges with dense network topologies, resulting in more complex node interactions during propagation. In Advogato, all algorithms demonstrate high influence ranges, with our algorithm reaching a peak of 3712. 415 at k=30, achieving approximately 1. 0%-1. 5%improvement over the ToupleGDD algorithm. In Public, our algorithm achieves an influence range of 6037. 037 at k=50, representing a 5. 9%enhancement compared to ToupleGDD. The narrowing performance gap in large datasets stems from the ceiling effect in network propagation. However, our algorithm maintains its performance advantage by leveraging global feature extraction through graph contrastive learning and dynamic decision-making via reinforcement learning, enabling it to identify superior seed combinations even in densely connected networks.

4.2.3 Comparative analysis of algorithm performance

Basic benchmark algorithms(Random and PageRank):Both algorithms demonstrate significantly inferior performance across all datasets compared to other optimization methods. The Random algorithm exhibits the weakest propagation capability in seed sets due to its non-targeted random selection mechanism. The PageRank algorithm, which solely evaluates global node importance without incorporating dynamic propagation characteristics, fails to effectively identify seed nodes with dissemination potential. When k=50, its influence coverage is merely 1-2 times that of the Random algorithm, far below other algorithms specifically designed for influence maximization. This indicates that global node importance does not equate to dissemination potential, and influence maximization algorithms must integrate propagation models with network topology features for optimal seed selection.

Traditional Greedy Algorithms vs. Deep Learning Algorithms(gIM, S2V-DQN):The gIM algorithm, as a classic greedy approach, demonstrates strong performance on certain datasets(e. g. , Advogato). However, it underperforms S2V-DQN and our proposed algorithm on datasets like Tv-show and Politician. This stems from cumulative errors in marginal gain calculations and the absence of collaborative propagation effects between seed nodes. The S2V-DQN algorithm, which integrates graph structure encoding with reinforcement learning, outperforms gIM across most datasets. Yet on topologically complex datasets(e. g. , Public), its performance gains are limited due to information loss in graph structure encoding. Additionally, training instability in some scenarios causes its performance growth to plateau after surpassing gIM at k=50.

The advanced heuristic algorithm(ToupleGDD)demonstrates superior performance across all benchmark algorithms, outperforming gIM and S2V-DQN across all datasets. This stems from its graph-based depth discounting approach, which effectively captures local propagation characteristics of neural networks and selectively identifies nodes with high local propagation capacity. However, as it relies solely on heuristic network topology features without global feature extraction or considering the serialized collaborative effects of seed selection, it consistently underperforms the proposed algorithm in all datasets.

Our algorithm achieves optimal expected impact range across all datasets and seed set sizes k, standing out as the only method that consistently outperforms other algorithms in all scenarios. Its core advantages are reflected in three key aspects:

The image comparison learning module, through self-supervised pre-training, extracts high-quality node propagation representations from network topology, edge weights, and node features. This effectively captures both global and local propagation potential of nodes, providing a reliable feature foundation for subsequent seed selection.

The problem of maximizing influence is transformed into a serialized decision-making process, and the dynamic collaborative selection of seed nodes is realized by combining Double DQN, which fully considers the complementary transmission between seed nodes, and avoids the marginal gain error of greedy algorithm and the feature limitation of heuristic algorithm.

The Monte Carlo-based marginal influence gain reward design achieves precise alignment between reinforcement learning strategies and the goal of maximizing influence, ensuring that the algorithm's training process consistently focuses on enhancing the propagation capability of the seed set, thereby guaranteeing the final decision-making performance.

4.3 Summary of Algorithm Performance Advantages

Experimental results from six real-world social network datasets demonstrate that our proposed collaborative decision-making algorithm, which combines graph-based contrastive learning and deep reinforcement learning, achieves significant performance advantages in influence maximization tasks. The key findings are summarized as follows:

First, it maintains performance leadership across social network datasets with varying topological scales(small, medium, and large), demonstrating no significant scenario limitations and suitability for diverse social network influence maximization tasks. Second, compared to the advanced ToupleGDD algorithm, it achieves performance improvements ranging from 1. 0%to 32. 4%across datasets, with the seed set exhibiting significantly broader expected influence ranges and enabling wider information dissemination. Finally, through serialized reinforcement learning decisions, it fully considers the collaborative propagation effects among seed nodes, avoiding the cumulative errors of traditional greedy algorithms and the feature singularity of heuristic methods. The seed set selection better aligns with the dynamic principles of information propagation.

The experimental results also validate the effectiveness of the graph contrastive learning pre-training+deep reinforcement learning decision framework in the influence maximization problem. Graph contrastive learning provides a reliable feature foundation for evaluating node propagation potential, while deep reinforcement learning offers an efficient decision-making method for collaborative selection of seed sets. The combination of these two approaches effectively addresses the shortcomings of traditional algorithms in feature extraction and decision optimization, providing a novel and effective solution to the influence maximization problem.

5 CONCLUSION

To address the limitations of traditional influence maximization algorithms in feature extraction and collaborative decision-making, this paper proposes a novel algorithm that integrates graph contrastive learning with deep reinforcement learning. By employing self-supervised pre-training to uncover node propagation characteristics, the algorithm transforms the problem into a sequential decision-making process for collaborative seed selection. Experiments on six real-world social networks validate the algorithm's effectiveness, achieving optimal propagation performance across various seed sizes. Compared to advanced heuristic algorithms, it demonstrates a performance improvement of 1. 0%to 32. 4%, while maintaining strong generalizability across different network scales. Future research could extend to dynamic social network scenarios by incorporating temporal features to optimize propagation models. Additionally, exploring lightweight network architectures to reduce computational complexity could enhance the algorithm's efficiency in ultra-large-scale social networks.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Cai Y, Dong S, Yuan S, et al. Network analysis models for variables and their applications. Advances in Psychological Science, 2020, 28(1): 178-190.
[2] Cui Xuelian, Narisa. Modeling of Online Word-of-Mouth Information Diffusion Based on Consumer Trust Relationship. Journal of Systems&Management, 2020, 29(6): 1090-1100.
[3] Wang Xuan, Zhang Yu, Zhou Junfeng, et al. Influence maximization algorithm based on social network. Journal on communications, 2022, 43(8): 151-163.
[4] Liu Zhiwei, Xia Zhiming. Semi-Linear Neural Networks Estimation of Partially Linear Model. Chinese Journal of Applied Probability and Statistics, 2023, 39(2): 218-238.
[5] Wang Jinghong, Wu Zhibing, Huang Peng, et al. Heterogeneous network representation learning based on metapath attribute fusion. JOURNAL OF SHANDONG UNIVERSITY(NATURAL SCIENCE), 2024, 59(3): 1-13.
[6] Wu Guodong, Zha Zhikang, Tu Lijing, et al. Research advances in graph neural network recommendation. CAAI Transactions on Intelligent Systems, 2020, 15(1): 14-24.
[7] Guan Fengxu, Zhang Hanyu, Lu Siqi, et al. Research status of diffusion models in computer vision. CAAI Transactions on Intelligent Systems, 2025, 20(2): 265-282.
[8] Wang XueSong, Wang RongRong, Cheng YuHu. A review of offline reinforcement learning based on representation learning. Acta Automatica Sinica, 2024, 50(6): 1104-1128
[9] Gu Xiaoqing, Yi Dangxiang, Liu Chunhe. Optimization of Topological Structure and Weight Value of Artificial Neural Network Using Genetic Algorithm. Journal of Guangdong University of Technology, 2006, 23(4): 64-69.
[10] Xu Yuefan, Xiao Wendong, Cao Zhengtao. Ensemble extreme learning machine approach for heartbeat classification by fusing 1d convolutional and handcrafted features. Chinese Journal of Engineering, 2021, 43(9): 1224-1232.
[11] Zou Xiaohong, Xu Chengwei, Chen Jing, et al. Research on seed node mining algorithm in large-scale temporal graph. Journal on communications, 2022, 43(9): 157-168.