

A node importance prediction algorithm based on graph attention and contrastive learning

Jun Ai(艾均), Yuming Zhang(张玉明), Zhan Su(苏湛), Chenye Guo(郭晨晔), and Mingsong Li(李铭松)
Citation: Chin. Phys. B, 2026, 35 (**5**): 058901. DOI: 10.1088/1674-1056/ae0680
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Six-degree gravity centrality for detecting influential nodes in networks

Jianbo Wang(王建波), Bohang Lin(林渤杭), Zhanwei Du(杜占玮), Ping Li(李平), and Xiao-Ke Xu(许小可)
Chin. Phys. B, 2025, 34 (**8**): 088902. DOI: 10.1088/1674-1056/adec62

Critical station identification of metro networks based on the integrated topological-functional algorithm: A case study of Chengdu

Zi-Qiang Zeng(曾自强), Sheng-Jie He(何圣洁), and Wang Tian(田旺)
Chin. Phys. B, 2025, 34 (**2**): 028902. DOI: 10.1088/1674-1056/ad9734

Dynamic modeling and analysis of brucellosis on metapopulation network: Heilongjiang as cases

Xin Pei(裴鑫), Xuan-Li Wu(武绚丽), Pei Pei(裴沛), Ming-Tao Li(李明涛), Juan Zhang(张娟), and Xiu-Xiu Zhan(詹秀秀)
Chin. Phys. B, 2025, 34 (**1**): 018904. DOI: 10.1088/1674-1056/ad92ff

AG-GATCN: A novel method for predicting essential proteins

Peishi Yang(杨培实), Pengli Lu(卢鹏丽), and Teng Zhang(张腾)
Chin. Phys. B, 2023, 32 (**5**): 058902. DOI: 10.1088/1674-1056/acb9f9

LCH: A local clustering H-index centrality measure for identifying and ranking influential nodes in complex networks

Gui-Qiong Xu(徐桂琼), Lei Meng(孟蕾), Deng-Qin Tu(涂登琴), and Ping-Le Yang(杨平乐)
Chin. Phys. B, 2021, 30 (**8**): 088901. DOI: 10.1088/1674-1056/abea86

A node importance prediction algorithm based on graph attention and contrastive learning

Jun Ai(艾均), Yuming Zhang(张玉明)†, Zhan Su(苏湛)‡,
Chenye Guo(郭晨晔), and Mingsong Li(李铭松)

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, *China*

(Received 12 August 2025; revised manuscript received 9 September 2025; accepted manuscript online 15 September 2025)

In complex network analysis, node ranking is vital for propagation prediction, structural optimization, and intervention strategy design, yet existing methods often fail to effectively integrate community information in dynamic settings. To address this, this paper proposes a node ranking method that combines graph attention mechanisms with contrastive learning. Community detection is employed to extract node-level community features, and a joint embedding module is designed to fuse global and local structures, thereby incorporating community information into node representations. Based on this, a multi-layer graph attention network adaptively learns node and neighborhood features, while contrastive learning mitigates interference from dynamic evolution and strengthens the model's ability to capture multi-scale structural differences. Experiments on multiple dynamic network datasets show that the proposed method significantly outperforms existing approaches in ranking accuracy, particularly in networks with higher average degrees and clearer community structures. These results validate the effectiveness of the method in enhancing feature representation and modeling multi-scale dynamic node influence.

Keywords: node importance, community features, graph attention, contrastive learning

PACS: 89.75.Hc, 89.20.Ff, 05.10.–a

DOI: 10.1088/1674-1056/ae0680 **CSTR:** 32038.14.CPB.ae0680

1. Introduction

Complex networks[1] serve as a fundamental framework for representing complex systems, which consist of a large number of interacting nodes and their connections. They are widely employed to model the intricate relationships among components within various systems. In recent years, complex networks have garnered significant attention due to their profound theoretical implications and broad applicability across diverse fields.

The objective of identifying influential nodes is to locate those that can exert a significant impact on the structure and function of a network, as well as those capable of rapidly disseminating information across large-scale systems based on node and network properties. This represents a fundamental problem in network science and holds considerable significance for a wide range of real-world applications.[2] For instance, in transportation networks, regularly maintaining roads that serve as critical hubs can prevent large-scale traffic breakdowns;[3] in gene regulatory networks, identifying genes responsible for disease development can accelerate medical treatment processes;[4] and in social networks, detecting influential bloggers who effectively reach a company's target audience can enhance the promotion of new products.[5]

Node importance identification methods have shown a clear evolution from shallow metrics based on structural features to deep models that integrate learning mechanisms. Traditional approaches primarily rely on local structural information of nodes, such as degree centrality (DC)[6] and local centrality (LC).[7] Although computationally efficient, these methods often struggle to accurately reflect a node's global influence within the network.[8] To address this limitation, subsequent studies incorporated macro-level structural factors, including betweenness centrality (BC),[9] closeness centrality (CC),[10] and eigenvector centrality (EC),[11] which leverage path and topological information to enhance global awareness. However, such methods often face scalability issues in large networks due to high computational costs. Community-based features were also introduced, such as the community-based centrality (CbC) method,[12] which emphasizes the roles of nodes within and across communities, though its effectiveness heavily depends on the accuracy of community detection.[13] With advancements in graph modeling, machine learning-based methods (ensemble machine learning, EML[14] and principal component analysis, PCA[15]) emerged, leveraging feature combinations and discriminative learning to model node importance. While more flexible, they often fail to capture higher-order dependencies in graph structures. In recent years, deep learning approaches have gained traction, especially models based on graph convolutional networks (GCNs), such as RCNN,[16] M-RCNN,[17] and LCNN,[18] which of-

†Corresponding author. E-mail: zym1013@163.com
‡Corresponding author. E-mail: suzhan@usst.edu.cn

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

fer powerful structural representation learning capabilities. Nonetheless, issues related to interpretability and generalizability across diverse networks remain to be addressed.

Despite recent advances, node importance ranking still faces key challenges. First, the diversity of network structures makes it difficult to jointly model micro-level centrality, macro topology, and community information.[16] Second, the dynamic nature of networks causes node importance to evolve over time, requiring effective modeling of temporal variation.[16] Third, deep models often lack interpretability and generalization, especially in dynamic networks.[17]

In response to these challenges, the proposed NGACL (a node importance prediction algorithm based on graph attention and contrastive learning) integrates community structure into node representations, applies graph attention to model multi-scale neighbor interactions, and employs contrastive learning to mitigate dynamic noise. The main contributions are summarized as follows.

(i) Community structure information is integrated into the node feature construction process, enhancing the model's capacity to capture multi-scale structural semantics and better characterize node importance.

(ii) A novel algorithm, NGACL, is developed. Through dynamic weighting of neighboring nodes and the combination of multilayer perceptrons with contrastive learning, the algorithm strengthens structural perception and improves the discriminative power of node embeddings for accurate importance prediction.

(iii) A joint optimization strategy is employed, incorporating a primary ranking loss alongside an auxiliary contrastive loss. This end-to-end training paradigm contributes to improved consistency and robustness in node importance ranking.

2. Related work

To address the challenges of multi-scale structural integration and dynamic evolution in node importance identification within complex networks, Louvain community detection is introduced in this study, combined with graph attention mechanisms and contrastive learning techniques, to explore node importance prediction methods. The Louvain algorithm[19] is a widely adopted and efficient method for community detection in complex networks. It partitions network structures into communities by optimizing modularity through a two-phase iterative process. In the first phase, nodes are assigned to neighboring communities that yield the highest modularity gain. In the second phase, a new network is constructed where each community becomes a single node, and the process is repeated until no further improvement in modularity is observed. Due to its low computational complexity and scalability, the Louvain algorithm is well suited for large-scale networks and has been extensively applied in various domains, including social and biological networks. Its main advantages include the ability to uncover hierarchical community structures, the absence of a requirement to predefine the number of communities, and efficiency in handling large datasets. Nonetheless, the algorithm is affected by the resolution limit of modularity, which can result in the merging of small but distinct communities. In addition, the detection results may vary between runs, particularly in networks with heterogeneous degree distributions or structural noise.

In this study, the Louvain algorithm is utilized to perform community partitioning. Based on the detected communities, node-level centrality features are computed within each community and normalized to form local structural descriptors. The analysis focuses on how community structures and centrality metrics can be jointly leveraged to better represent the positional importance of nodes within their respective communities. This integration aims to improve the accuracy of node importance.

The graph attention network (GAT)[20] is a deep neural network architecture designed to process graph-structured data. By incorporating an attention mechanism into the message-passing process of traditional graph neural networks, GAT adaptively assigns weights to neighboring nodes based on their structural relevance. Unlike approaches that aggregate neighbor features with fixed or uniform weights, GAT enables the capture of fine-grained local structural variations, thereby improving the expressiveness and discriminative capability of node representations. In recent years, GAT has been extensively applied in graph-centric tasks across diverse domains, such as social network analysis, recommender systems, bioinformatics, and traffic forecasting,[21–23] and has become a prominent architecture within the field of graph neural networks. However, existing GAT-based methods rarely explicitly integrate community-aware features, which can limit their ability to capture multi-scale structural semantics and global network context.

In this study, GAT is adopted to model the nonlinear relationship between node features and their importance scores in the information propagation process. The attention mechanism allows for the modeling of heterogeneous interactions between nodes and their neighbors, facilitating more accurate estimation of node importance.

Contrastive learning (CL) has gained prominence as a self-supervised representation learning paradigm, which learns effective feature embeddings by distinguishing between positive and negative sample pairs. The core objective is to maximize the agreement between positive pairs while minimizing that between negative ones, thereby facilitating the extraction of latent structural patterns under unsupervised or weakly supervised settings. The conceptual foundation of CL

can be traced back to noise contrastive estimation (NCE),[24] where unnormalized probability distributions were estimated by differentiating between real and artificially generated noise samples. This formulation laid the theoretical groundwork for subsequent contrastive objectives. In 2018, contrastive predictive coding (CPC)[25] was proposed to maximize the mutual information between local and global representations via the InfoNCE loss. This significantly improved the quality and generalizability of learned representations, accelerating the adoption of contrastive learning across various domains. Substantial progress has since been reported in areas such as computer vision, natural language processing, recommender systems, and graph representation learning.[26,27] By constructing informative positive and negative samples, contrastive learning has been shown to enhance the model's ability to capture deep semantic and structural relationships, leading to improved robustness and downstream task performance.

In this study, contrastive learning is integrated into the training framework as an auxiliary supervisory signal, complementing the primary loss function. This integration guides the model to capture more fine-grained feature distinctions among nodes. Moreover, the potential of contrastive objectives to improve the understanding of both local neighborhoods and global structural roles is explored, with the aim of enhancing the accuracy of importance estimation across complex networks.

3. Methodology

3.1. Problem definition

Node importance is a fundamental metric for quantifying the significance of individual nodes within a network topology. It reflects each node's potential impact on dynamic processes such as information diffusion or epidemic spreading. The formal definition is given as follows.

Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ represents the set of edges. The objective of node importance prediction is to learn a function $f : \boldsymbol{x}_v \to \hat{y}_v$, which maps the feature vector $\boldsymbol{x}_v$ of each node $v \in \mathcal{V}$ to its predicted importance score $\hat{y}_v \in \mathbb{R}$, thereby approximating its ground-truth label y_v. The node feature vector $\boldsymbol{x}_v$ typically incorporates topological descriptors such as structural centralities, community affiliation indicators, and other node-level attributes. Ground-truth importance labels y_v can be obtained through epidemic simulation models (*e.g.*, susceptible-infected-recovered, SIR),[28] classical centrality metrics, observational data, or expert annotations. In this study, feature construction is based on a combination of structural centrality measures and community detection results, while importance labels are generated using the SIR model, serving as supervisory signals for model training.

The task is framed as a regression problem, wherein a nonlinear mapping is established between the structural properties of each node and its influence observed during simulated spreading dynamics. This formulation enables the prediction of importance scores $\hat{y}_v$ that reflect the relative impact of nodes within the network under dynamic processes.

3.2. Algorithm framework

The overall framework of NGACL is illustrated in Fig. 1. The architecture consists of five core modules: the node feature extraction module, the GAT encoding module, the MLP prediction module, the CL module, and the loss function module. A joint training strategy integrates the main and auxiliary tasks for model optimization.

In the node feature extraction module, multiple structural features are extracted from the network for each node. These include global centrality metrics as well as local centrality features within the node's community. All features are normalized and concatenated to form the input feature vector $\boldsymbol{x}_v$, providing structural context for subsequent processing.

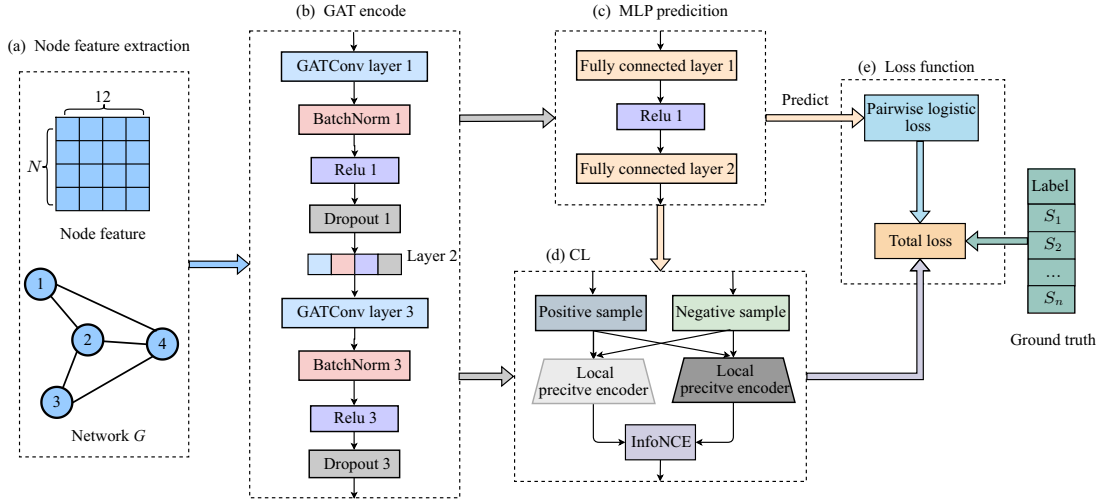


Fig. 1. An overview of the NGACL framework.

In the GAT encoding module, a three-layer GAT encodes structural information. Given node features $\boldsymbol{x}_v$ and the network topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the attention mechanism estimates the importance of neighboring nodes. The first two layers use multi-head attention, while the third adopts a single-head mechanism. Each layer is followed by batch normalization, a ReLU activation function, and Dropout to enhance stability and generalization. The final output is the node embedding $\boldsymbol{z}_v$.

In the MLP prediction module, the node embedding $\boldsymbol{z}_v$ is passed through a two-layer MLP for regression, yielding the predicted node importance score $\hat{y}_v$. This stage applies nonlinear transformations to approximate influence scores.

In the CL module, positive and negative node pairs are constructed based on predicted scores and structural features. A contrastive learning loss is computed to encourage nodes with similar structures and scores to have closer embeddings, while separating dissimilar ones. This improves the discriminative capacity of the embedding space.

The loss function module adopts a joint optimization approach. The main task employs a pairwise ranking loss that penalizes incorrect orderings of node pairs based on ground-truth importance. The auxiliary task applies the contrastive loss. The overall objective $\mathcal{L}$ is a weighted combination of both, enabling end-to-end training via backpropagation and enhancing ranking consistency.

Through the coordinated operation of these modules, NGACL effectively represents the features of nodes in the network and enables accurate prediction of their spreading capabilities.

From a physics perspective, NGACL can be interpreted through analogies with critical phenomena in complex systems. Community structures act like mesoscopic order parameters capturing intermediate-scale organization, while the graph attention mechanism resembles interaction weighting near a critical point. The contrastive learning module plays a role similar to renormalization, filtering noise while retaining essential structural signals.

The training and prediction process is outlined in Algorithm 1.

Algorithm 1 Node Importance Prediction Algorithm

Require: Network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, node features $\boldsymbol{x}_v$, labels y_v, training ratio $r = 0.7$, number of epochs N
Ensure: Predicted node importance scores $\hat{y}_v$
Feature Processing: For each node $v \in \mathcal{V}$, compute structural features and normalize to obtain input vector $\boldsymbol{x}_v$
Data Splitting: Divide the dataset into training set $\mathcal{T}_{\text{train}}$ and test set $\mathcal{T}_{\text{test}}$ by timestamp
for *epoch* $= 1$ to N **do**
    $\boldsymbol{z}_v = \text{GAT}(\boldsymbol{x}_v, \mathcal{T}_{\text{train}})$ ▷ Extract node embeddings
    $\hat{y}_v = \text{MLP Predictition}(\boldsymbol{z}_v)$ ▷ Predict node importance
    $\mathcal{L}_{\text{rank}}$ ▷ Compute pairwise ranking loss
    $\mathcal{L}_{\text{contrastive}}$ ▷ Compute contrastive loss
    $\mathcal{L} = \mathcal{L}_{\text{rank}} + \lambda \mathcal{L}_{\text{contrastive}}$ ▷ Compute total loss
    Backpropagate and update model parameters
    Save model
end for
Testing: Predict $\hat{y}_v$ on the test set $\mathcal{T}_{\text{test}}$
return $\hat{y}_v$

3.3. Node feature extraction

Node features are a crucial component of the algorithm, as they directly affect the final prediction performance. The features employed in this study include degree centrality, represented as first-order degree D_1, second-order degree D_2, and third-order degree D_3, along with the H-index, represented as first-order H-index H_1, second-order H-index H_2, and third-order H-index H_3. Each type of feature is calculated at two distinct scales: the global scale and the community scale. This yields four feature vectors: D_v^{network}, $D_v^{\text{community}}$, H_v^{network}, and $H_v^{\text{community}}$, which capture the structural characteristics of a node from different hierarchical perspectives. These vectors are concatenated along the feature dimension to construct the input feature vector for each node, denoted as $\boldsymbol{x}_v$. The following section describes the computation of these features under both the global and community scales in detail.

3.3.1. Global node feature learning

Figure 2 illustrates the construction process of node features at the global network scale. To capture the structural information of network nodes across different neighborhood ranges, multi-order degree centrality and H-index features are introduced to characterize node influence in complex net-

works. For each node v, the first-order degree centrality and H-index are first computed in the original network. These features are then recursively aggregated from neighboring nodes to derive the second- and third-order features. This hierarchical aggregation facilitates the incorporation of multi-hop neighborhood information. All multi-order feature vectors are subsequently concatenated after applying Min–Max normalization, resulting in the final structural feature representation of each node at the global scale. This representation is denoted as D_v^{network} and H_v^{network}, providing a multi-scale, information-rich input for downstream node embedding and importance prediction tasks.

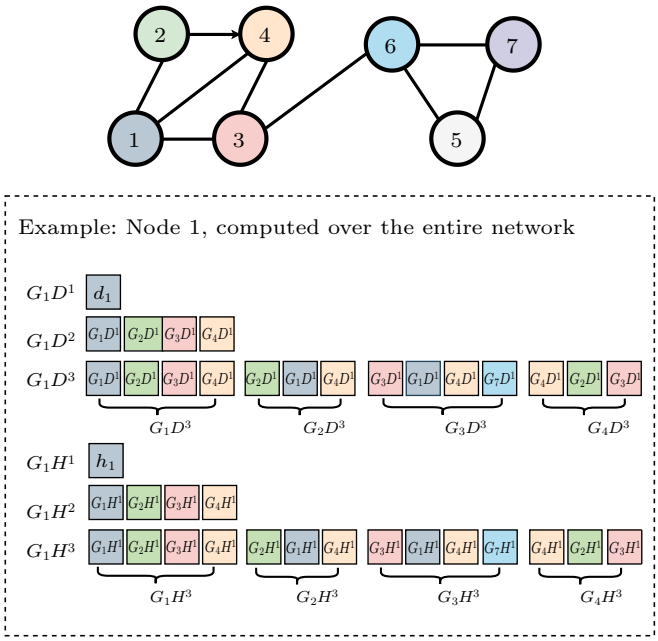


Fig. 2. Global node feature extraction.

Each node is initially assigned global first-order features, including degree centrality and H-index, as shown below:

$$G_v^{D_1} = \deg(v), \tag{1}$$

$$G_v^{H_1} = \max\{h \,|\, |\{u \in N(v) \mid \deg(u) \geq h\}| \geq h\}. \tag{2}$$

Then, the second-order and third-order features are generated recursively. The second-order feature is composed of the node's own first-order feature and the sum of its neighbors' first-order features, as shown below:

$$G_v^{D_2} = G_v^{D_1} + \sum_{u \in N(v)} G_u^{D_1}, \tag{3}$$

$$G_v^{H_2} = G_v^{H_1} + \sum_{u \in N(v)} G_u^{H_1}. \tag{4}$$

Similarly, the third-order features are computed in the same recursive manner, by aggregating the node's second-order feature with those of its neighbors, as shown below:

$$G_v^{D_3} = G_v^{D_2} + \sum_{u \in N(v)} G_u^{D_2}, \tag{5}$$

$$G_v^{H_3} = G_v^{H_2} + \sum_{u \in N(v)} G_u^{H_2}. \tag{6}$$

Finally, the third-order features are concatenated and normalized using Min–Max scaling to obtain the global feature representation of each node, as shown below:

$$D_v^{\text{network}} = \text{MinMax}\left(\left[G_v^{D_1}, G_v^{D_2}, G_v^{D_3}\right]\right), \tag{7}$$

$$H_v^{\text{network}} = \text{MinMax}\left(\left[G_v^{H_1}, G_v^{H_2}, G_v^{H_3}\right]\right). \tag{8}$$

3.3.2. Community node feature learning

Figure 3 illustrates the construction process of node features at the community scale. To better capture the local structural characteristics of nodes within their communities, community-based feature modeling is introduced as a complement to global network features. Let the community to which node v belongs be denoted by C, and let the set of its intra-community neighbors be $N_C(v)$. At the community level, two types of recursive features are constructed: the degree centrality-based feature $D_v^{\text{community}}$ and the H-index-based feature $H_v^{\text{community}}$. These features are aggregated up to the third order to capture local and semi-local structural semantics within communities.

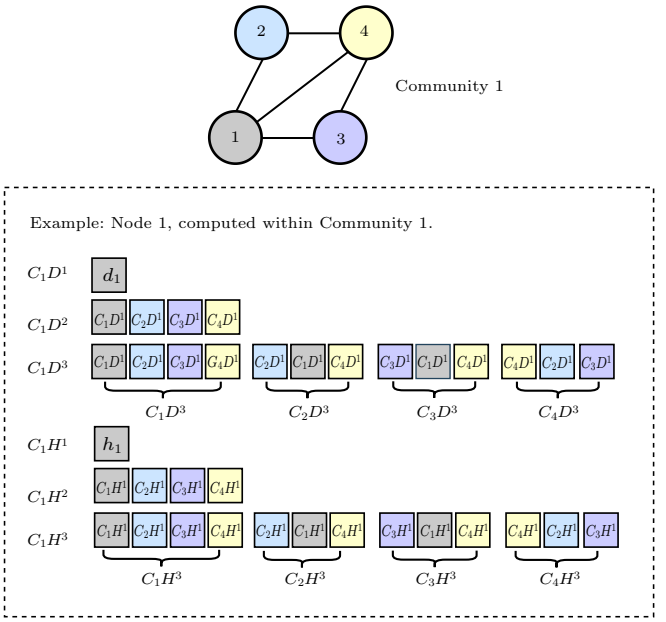


Fig. 3. Community node feature extraction.

The first-order community features for each node are initialized using degree centrality and H-index, as defined below:

$$C_v^{D_1} = \deg_C(v), \tag{9}$$

$$C_v^{H_1} = \max\{h \,|\, |\{u \in N_C(v) \mid \deg_C(u) \geq h\}| \geq h\}. \tag{10}$$

Then, we recursively generate the second-order and third-order structural features. The second-order features consist of the node's first-order community features and the sum of its neighbors' first-order features, as shown below:

$$C_v^{D_2} = C_v^{D_1} + \sum_{u \in N_C(v)} C_u^{D_1}, \tag{11}$$

$$C_v^{H_2} = C_v^{H_1} + \sum_{u \in N_C(v)} C_u^{H_1}. \tag{12}$$

Similarly, the third-order features are calculated by aggregating the node's second-order features and those of its neighbors, as shown below:

$$C_v^{D_3} = C_v^{D_2} + \sum_{u \in N_C(v)} C_u^{D_2}, \tag{13}$$

$$C_v^{H_3} = C_v^{H_2} + \sum_{u \in N_C(v)} C_u^{H_2}. \tag{14}$$

Finally, the three-order features are concatenated and normalized using Min–Max normalization to obtain the final community-based structural feature representation of the node, as shown below:

$$D_v^{\text{community}} = \text{MinMax}\left(\left[C_v^{D_1}, C_v^{D_2}, C_v^{D_3}\right]\right), \tag{15}$$

$$H_v^{\text{community}} = \text{MinMax}\left(\left[C_v^{H_1}, C_v^{H_2}, C_v^{H_3}\right]\right). \tag{16}$$

3.4. Loss function design

The loss function in this paper consists of a main task loss and a contrastive learning loss. The main task adopts a pairwise ranking loss to optimize the consistency of the model's node importance prediction. Let the predicted scores be $\hat{\boldsymbol{y}}$, and the ground-truth scores be $\boldsymbol{y}$. For a node pair (i, j), the true score difference is $\Delta y_{ij} = y_i - y_j$, and the predicted score difference is $\Delta \hat{y}_{ij} = \hat{y}_i - \hat{y}_j$. The pairwise label is defined as

$$S_{ij} = \mathbb{I}(y_i > y_j). \tag{17}$$

We use $\Delta \hat{y}_{ij}$ as the logits and apply the binary cross entropy with logits loss to compute the pairwise ranking loss

$$\mathcal{L}\text{rank} = \frac{1}{|M|} \sum (i, j) \in M\text{BCE}(\Delta \hat{y}ij, Sij), \tag{18}$$

where M denotes the set of node pairs satisfying the masking constraint.

Additionally, we introduce a contrastive learning loss. For each node embedding $\boldsymbol{z}_i$, a predicted importance score $\hat{y}_i$ is generated by a predictor $\text{Dec}(\cdot)$, and concatenated with $\boldsymbol{z}_i$ to form a positive sample pair

$$\boldsymbol{p}_i^+ = [\hat{y}_i | \boldsymbol{z}_i] \tag{19}$$

The positive sample is mapped by a local encoder and a global encoder to produce z_i^{local} and z_i^{global}, respectively, and their cosine similarity is computed as

$$s_i^+ = \cos(z_i^{\text{local}}, z_i^{\text{global}}). \tag{20}$$

By shuffling the index of $\boldsymbol{z}_i$, we construct a negative sample pair $\boldsymbol{p}_i^-$ and compute the corresponding negative similarity s_i^-. The InfoNCE contrastive loss is defined as

$$\mathcal{L}\text{con} = \frac{1}{N} \sum i = 1^N \text{CE}([s_i^+, s_i^-], y_i = 0), \tag{21}$$

where CE denotes the cross-entropy loss function, and the label $y_i = 0$ indicates that the positive sample is the first element.

The overall loss combines the main ranking loss and the contrastive loss with a weighting factor λ, which controls the influence of contrastive learning on the overall optimization. This design aims to enhance the discriminability of learned representations while preserving the performance of the main task

$$\mathcal{L} = \mathcal{L}\text{rank} + \lambda \cdot \mathcal{L}\text{con}. \tag{22}$$

3.5. Label generation

In this study, node importance prediction is formulated as a regression task, where each node is assigned a numerical label. To generate supervisory signals, the classical susceptible-infected-recovered SIR diffusion model[28] is employed to simulate the spreading capability of nodes within the network.

The SIR model classifies nodes into three states: susceptible, infected, and recovered. At each propagation step, infected nodes transmit the infection to their neighbors with an infection probability β, while recovering with a recovery probability γ. During the diffusion initialization, only the target node is set as infected, and all other nodes are susceptible. The process continues until no infected nodes remain, reaching a steady state.

For each simulation, the total number of nodes eventually infected or recovered is considered as the spreading influence of the target node. In this work, the recovery probability is fixed at $\gamma = 1$, and the infection probability β is controlled by a tunable parameter α. To improve estimation stability, each node undergoes 1000 independent simulations, and the average diffusion scale is taken as its spreading influence label.[29]

4. Experiments

4.1. Datasets

To comprehensively evaluate the performance of the proposed algorithm, three synthetic networks and five real-world networks are selected, covering social networks, research collaboration networks, email networks, protein interaction networks, and power grid networks.

The real-world networks used in the experiments are briefly described as follows:

(i) Enron:[30] This dataset records the email communication network among employees of the Enron Corporation. Nodes represent employees, and edges correspond to email exchanges between colleagues.

(ii) COLAB:[31] This dataset contains collaboration data among 315 authors. Nodes represent authors, and edges indicate co-authorship relationships.

(iii) Facebook:[32] This dataset captures social connections among users on the Facebook platform, including friend-

ships and interactions, enabling the study of social dynamics, information diffusion, and community structure.

(iv) DBLP-3 and DBLP-5:[33] Extracted from the DBLP computer science bibliography, these datasets represent authors as nodes and co-authorships as edges. Authors are clustered into three and five research areas, respectively, with static temporal partitions.

The synthetic networks employed in the experiments include:

(I) Barabási–Albert (BA):[34] A typical scale-free network model. Synthetic BA networks with an average degree of 3 are constructed, and datasets comprising 10 temporal snapshots are generated.

(II) Merging and splitting (Merge): At each time step, five communities are split into smaller ones, while another five are merged pairwise to form new communities. These networks are synthesized using a dynamic benchmark generator[35,36] that produces unweighted, undirected, temporally evolving graphs with embedded ground-truth community structures. Each network contains 10 temporal snapshots, each with 1000 nodes, average degree 20, and maximum degree 50. The mixing parameter is set to $\mu = 0.2$, controlling the proportion of edges connecting nodes across communities.

(III) Expansion and contraction (Expansion): At each time step, five communities are randomly selected and their sizes are expanded or contracted by 25%. These datasets share the same synthesis procedure and network properties as described for Merge.

The statistical properties of these datasets are summarized in Table 1, including the number of nodes, number of edges, average degree, and number of communities.

Each dataset is divided into training and test sets with a ratio of 7:3. The training set is used for model learning and parameter optimization, while the test set serves to evaluate the model's performance on unseen data.

Table 1. Basic statistics of each network dataset.

Dataset	Timestamps	Nodes	Edges	Avg. degree	Communities
Enron	8, 9, 10	119, 111, 116	245, 238, 266	4.12, 4.29, 4.59	11, 10, 8
COLAB	7, 8, 9	249, 229, 244	287, 249, 308	2.31, 2.18, 2.53	47, 39, 36
Facebook	7, 8	592, 552	1333, 1068	4.50, 3.87	26, 22
BA	7, 8, 9	707, 662, 598	1453, 1316, 1108	4.11, 3.98, 3.71	19, 20, 17
Expansion	7, 8, 9	985, 964, 973	8953, 8681, 8806	18.18, 18.01, 18.10	32, 32, 32
Merge	7, 8, 9	1000, 1000, 1000	8562, 8723, 8664	17.12, 17.45, 17.33	29, 31, 33
DBLP3	7, 8, 9	1289, 1226, 1071	1308, 1243, 1142	2.03, 2.03, 2.13	355, 321, 296
DBLP5	7, 8, 9	2203, 2141, 1790	2356, 2343, 2016	2.14, 2.19, 2.25	561, 513, 455

4.2. Evaluation metrics

To assess the accuracy of node importance rankings, this study employs the Kendall's τ correlation coefficient[37] as the evaluation metric, which quantifies the consistency between the predicted node ranking and the ground-truth ranking. Kendall's τ is a non-parametric statistic that measures the ordinal association between two sequences and is commonly applied in influence maximization tasks.

Suppose two ranking lists A and B, each of length n, a pair of elements (i, j) is defined as *concordant* if their relative ordering is the same in both lists (i.e., $A_i > A_j$ and $B_i > B_j$, or $A_i < A_j$ and $B_i < B_j$); otherwise, it is defined as *discordant*.

The Kendall's τ coefficient is computed by

$$\tau = \frac{n_{\mathrm{c}} - n_{\mathrm{d}}}{\frac{1}{2}n(n-1)}, \tag{23}$$

where n_{c} denotes the number of concordant pairs, n_{d} denotes the number of discordant pairs, and n is the total number of elements in the ranking list. The value of Kendall's τ ranges from -1 to 1: a value close to 1 indicates strong agreement between the two rankings, a value close to -1 indicates complete disagreement, and $\tau = 0$ indicates no significant correlation.

4.3. Baselines

This study considers both centrality-based and deep learning-based baseline algorithms to assess node importance. The selected baseline methods are summarized as follows:

(i) Degree centrality[6] (DC): Measures node importance by counting the number of its first-order neighbors.

(ii) K-core centrality[38] (KC): Reflects a node's structural position by iteratively pruning nodes with degrees less than k, thus stratifying the network into hierarchical layers.

(iii) Betweenness centrality[9] (BC): Quantifies node importance based on the extent to which a node lies on shortest paths between other nodes, highlighting its role in connecting different parts of the network.

(iv) Closeness centrality[6] (CC): Defines node importance as the inverse of the average shortest path length to all other nodes; nodes with smaller average distances are considered more central.

(v) Eigenvector centrality[39] (EC): Assigns importance recursively, assuming that connections to high-scoring nodes contribute more to a node's own score.

(vi) CAGM:[40] An adaptive gravity-based model that integrates node communicability, influence radius, and local structural properties to estimate node importance.

(vii) LCNN:[29] A convolutional neural network-based model that captures local structural features through multi-scale indicators to predict node influence in complex networks.

All comparative experiments are conducted on dynamic network datasets. Given the limited availability of methods specifically designed for dynamic settings, this study primarily includes LCNN and CAGM as the key reference baselines. For LCNN, the results are obtained using the optimal parameter configurations reported in the original work.

5. Results

5.1. Evaluation results

As shown in Table 2, node importance prediction performance is evaluated using Kendall's τ, with the best results highlighted in bold and the second-best underlined. The experimental results indicate that NGACL outperforms other algorithms in the vast majority of cases. Only at a few timestamps in the Enron and Facebook datasets does CAGM slightly surpass it, but the performance difference remains within 4%. On average, NGACL achieves approximately 40% improvement over CAGM across all datasets. Specifically, on the COLAB dataset, the improvement ranges from 24% to 41%; on the DBLP3 dataset, from 14% to 27%; and on datasets with prominent community structures such as Expansion and Merge, the improvement can exceed 100%.

Figures 4 and 5 present a performance comparison of ranking lists generated by various algorithms against those produced by the SIR model, evaluated using Kendall's τ correlation coefficient. In these evaluations, the horizontal axis represents the adjustable infection probability parameter $\alpha \in [1, 1.9]$, while the vertical axis shows the corresponding Kendall's τ values. Specifically, figure 4 illustrates the results on synthetic networks, and figure 5 presents those on real-world networks.

As shown in Figs. 4 and 5, NGACL consistently outperforms LCNN and various centrality-based methods across both synthetic and real-world networks, and achieves better performance than CAGM on most datasets. In particular, on the Expansion and Merge datasets, NGACL yields nearly twice the Kendall's τ values compared to CAGM. On the DBLP3 and COLAB datasets, it also demonstrates clear advantages. For the DBLP5 dataset, NGACL similarly surpasses CAGM in most settings. On the BA dataset, NGACL performs slightly better than CAGM, while on the Enron and Facebook datasets, their performances are comparable. This may be due to the relatively small number of communities and less distinct local spreading structures in these networks, which limit the effectiveness of feature-based methods.

As shown in Fig. 6, the vertical axis adopts a logarithmic scale to better visualize the differences in running time among algorithms. The values atop the bars indicate the actual execution time for each algorithm. Except for the Enron and COLAB datasets, the running time of NGACL is markedly lower than that of CAGM, suggesting greater computational efficiency in most cases. While CAGM performs adequately on networks with fewer nodes and edges, NGACL exhibits more consistent and efficient performance across all datasets.

Table 2. Average Kendall's τ values for each dataset.

Dataset	Timestamp	BC	CC	DC	EC	KS	LCNN	CAGM	NGACL
BA	7	0.1193	0.1123	0.1338	0.1127	0.0943	0.1255	0.7942	**0.8082**
BA	8	0.1562	0.1245	0.1588	0.1124	0.1424	0.1279	0.7902	**0.8052**
BA	9	0.1320	0.1126	0.1352	0.1026	0.1302	0.1214	0.7787	**0.8154**
COLAB	7	–0.0398	–0.0251	–0.0525	-0.0112	–0.0571	–0.0434	0.6661	**0.8284**
COLAB	8	0.0529	0.0155	0.0673	0.0254	0.0024	0.0048	0.6192	**0.8360**
COLAB	9	0.0402	–0.0250	0.0313	-0.0286	–0.0081	–0.0250	0.5756	**0.8109**
DBLP3	7	–0.0071	0.0088	-0.0041	0.0111	–0.0068	0.0109	0.6750	**0.8420**
DBLP3	8	0.0003	–0.0057	-0.0047	–0.0025	–0.0077	0.0176	0.6781	**0.8366**
DBLP3	9	0.0471	0.0489	0.0467	0.0492	0.0514	0.0251	0.7083	**0.8083**
DBLP5	7	–0.0255	–0.0080	–0.0081	–0.0037	–0.0106	0.0052	0.6736	**0.8584**
DBLP5	8	0.0066	–0.0061	0.0067	–0.0073	0.0114	0.0247	0.8445	**0.8679**
DBLP5	9	–0.0423	–0.0042	–0.0120	-0.0026	–0.0089	–0.0222	0.6986	**0.8577**
Enron	8	0.0383	–0.0013	0.0246	–0.0020	0.0212	0.0188	0.7653	**0.8131**
Enron	9	0.1270	0.0584	0.0689	0.0375	0.0252	0.0295	**0.8123**	0.8001
Enron	10	0.0181	0.0439	0.0446	0.0547	0.0317	0.0487	**0.7709**	0.7444
Expansion	7	0.0746	0.0428	0.0943	0.0037	0.0726	0.0625	0.4186	**0.8698**
Expansion	8	0.0570	0.0335	0.0763	0.0219	0.0636	0.0509	0.3882	**0.8364**
Expansion	9	0.0981	0.0446	0.1073	0.0503	0.0925	0.0780	0.4340	**0.8626**
Facebook	7	0.0572	0.0894	0.0731	0.0794	0.0962	0.0834	**0.8355**	0.8054
Facebook	8	–0.0645	–0.0072	–0.0279	0.0086	0.0069	–0.0152	**0.7578**	0.7929
Merge	7	–0.0115	–0.0063	–0.0065	–0.0137	–0.0086	–0.0165	0.4106	**0.8807**
Merge	8	–0.0049	–0.0036	0.0104	–0.0011	0.0114	0.0034	0.3511	**0.8711**
Merge	9	0.0397	0.0207	0.0402	-0.0088	0.0222	0.0098	0.4076	**0.8486**

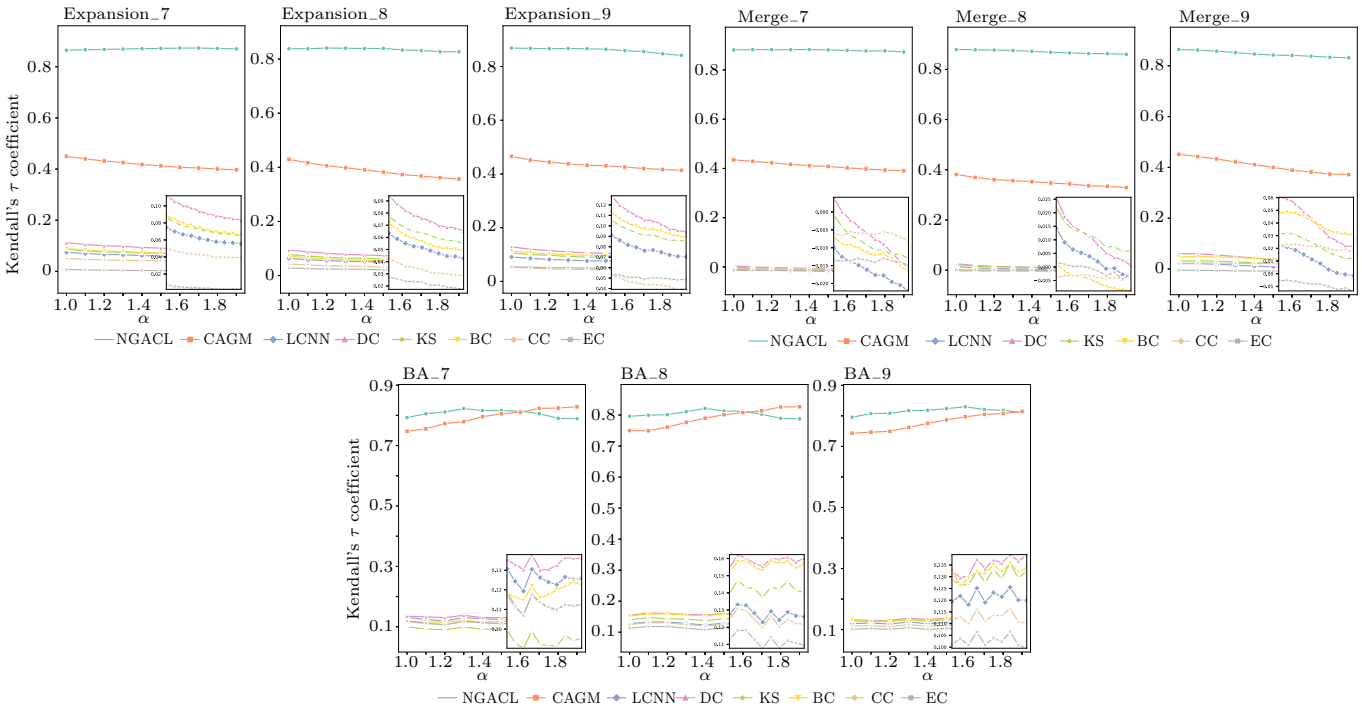


Fig. 4. Synthetic networks: Kendall's τ analysis.

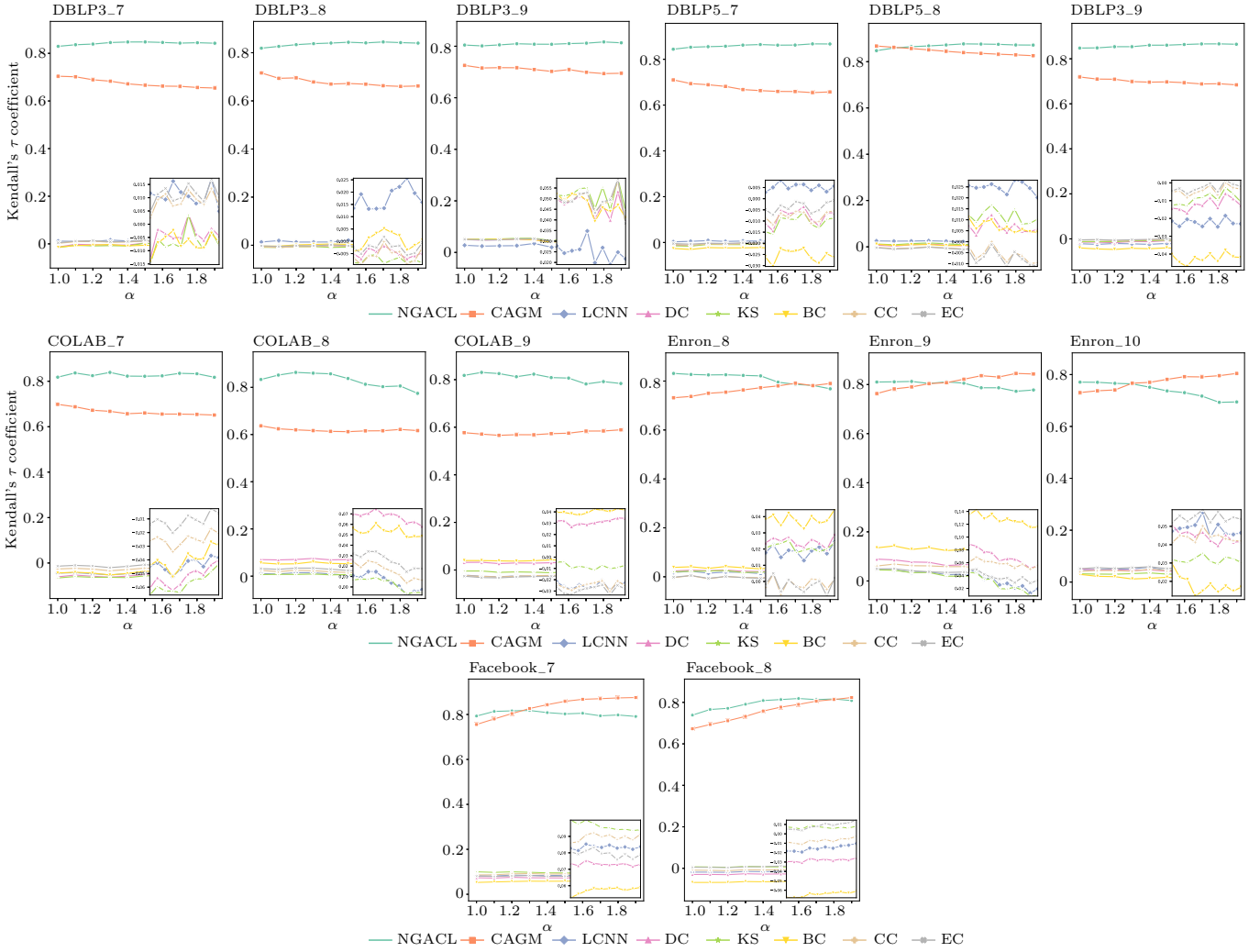


Fig. 5. Real-world networks: Kendall's τ analysis.

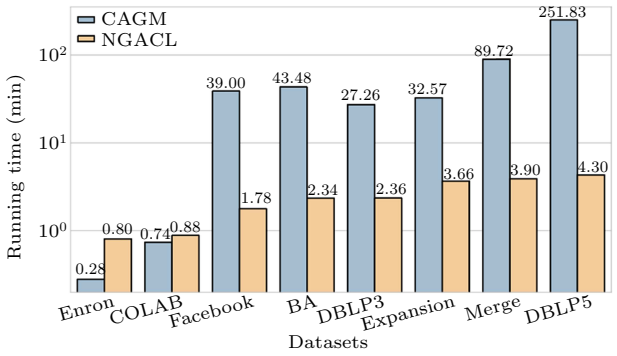


Fig. 6. Comparison of running time between CAGM and NGACL.

In addition, CAGM requires substantially more hardware resources, with experiments necessitating a server equipped with 60 GB of memory. In contrast, NGACL completes all experiments on a standard workstation with only 16 GB of memory, highlighting its advantage in both time and space complexity.

5.2. Hyperparameter and ablation experiments

To assess the influence of key hyperparameters on model performance and to examine the effectiveness of the contrastive learning module, a series of hyperparameter tuning and ablation experiments were conducted. These experiments aim to analyze the contribution of each component to the overall predictive capability.

A systematic evaluation was carried out to investigate the effect of the learning rate on model performance, focusing on how different values influence convergence and prediction accuracy. As a critical parameter in optimization, the learning rate controls the step size during parameter updates and thus impacts both training stability and final performance. An appropriate learning rate is essential to avoid divergence or excessively slow convergence.

The model's Kendall's τ performance on synthetic networks under different learning rates is illustrated in Fig. 7(a), while results on real-world networks are shown in Fig. 7(b). Overall, the optimal performance across various network types was achieved with a learning rate of 0.05. Across all datasets, performance decreases when the learning rate is below 0.05, and notably, a learning rate of 0.005 yields better overall results than 0.01.

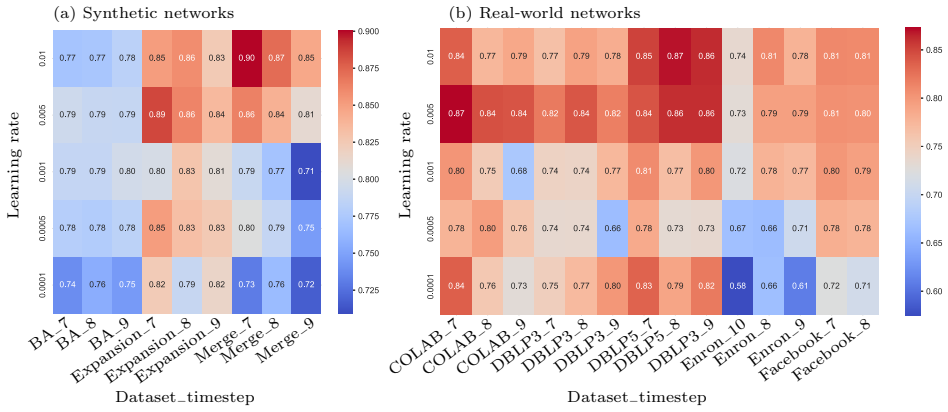


Fig. 7. Learning rate analysis on synthetic (a) and real-world (b) networks.

This study focuses on predicting node importance in complex networks. To improve the discriminative capability of node representations, the InfoNCE loss function based on contrastive learning was incorporated into the model, with a hyperparameter λ introduced to regulate its influence. A series of ablation experiments were conducted to evaluate the contribution of this loss term to prediction performance.

The effect of different λ values on model performance was first examined. As shown in Figs. 8(a) and 8(b), results across multiple datasets indicate that when $\lambda = 0.5$, the model achieves the most balanced performance, demonstrating its best generalization capability. When $\lambda = 0.25$, the model also performs relatively consistently, but the overall results are inferior to those at 0.5. In contrast, when $\lambda = 0.75$ or 0.1, some datasets exhibit Kendall's τ below 0.7, indicating poor generalization. Consequently, λ was fixed at 0.5 for all subsequent experiments to ensure stable performance.

Moreover, figures 9(a) and 9(b) compare the model performance with and without the contrastive learning module. Specifically, (i) GAT: model based on the proposed method without the contrastive learning module, excluding the InfoNCE loss. (ii) GAT+CL: model including the contrastive learning module, which corresponds to the proposed NGACL.

The results demonstrate that integrating the InfoNCE loss substantially improves median Kendall's τ on synthetic networks such as Expansion and Merge, and yields consistent gains across all real-world datasets.

In summary, the InfoNCE loss enhances the model's ability to capture complex nonlinear relationships between node spreading capability and graph structure, thereby improving the accuracy of node importance prediction.

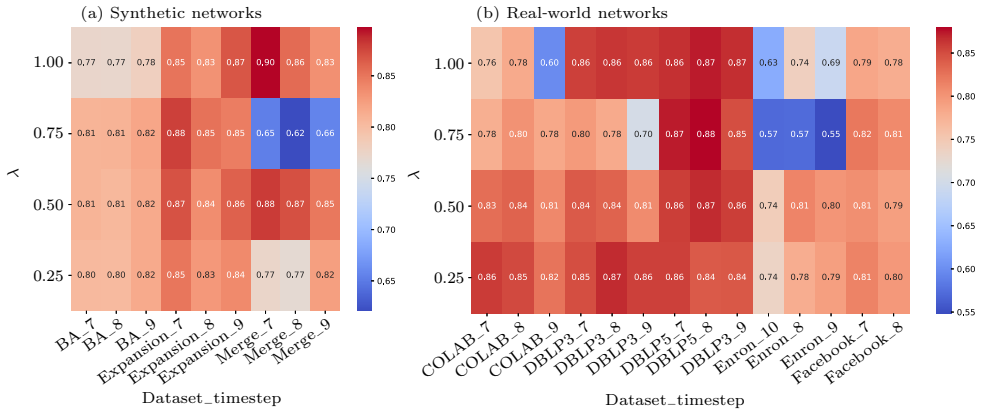


Fig. 8. λ parameter analysis on (a) synthetic and (b) real-world networks.

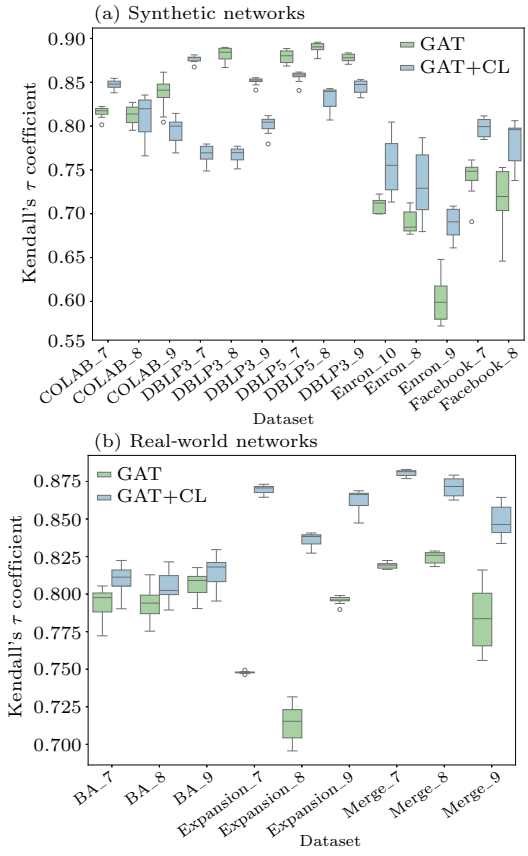


Fig. 9. Ablation experiment results analysis on synthetic and real-world networks.

5.3. Results analysis

Overall, the NGACL algorithm achieves the best performance in most cases, with only slight advantages observed for the CAGM algorithm at certain timestamps on the Enron and Facebook datasets. Even in these instances, NGACL maintains a high level of performance, demonstrating strong robustness.

In synthetic networks characterized by a higher number of communities and pronounced local clustering, NGACL exhibits superior predictive capabilities. Notably, on the Expansion and Merge datasets, NGACL's Kendall's τ values significantly exceed those of other baseline methods. From a physics perspective, this behavior is analogous to phase transitions in complex systems, where small structural changes can trigger large-scale shifts in system behavior. The incorporation of community-based features allows NGACL to capture these critical-like transitions, leading to superior predictive performance on networks with pronounced clustering. This improvement can be attributed to the integration of community-based node features during the feature construction phase, combined with the GAT module's effective focus on key neighboring nodes, which together enhance the representation of node spreading influence.

NGACL also shows stable performance on real-world networks. In large-scale datasets with clear community structures, such as DBLP3, DBLP5, and COLAB, NGACL consistently outperforms CAGM and traditional centrality measures. This advantage stems from the incorporation of community detection mechanisms in feature construction, providing higher-level structural semantic information for node embeddings. Additionally, the graph attention mechanism effectively captures multi-scale structural features, highlighting important nodes situated within critical communities.

On datasets like Enron and Facebook, where community numbers are relatively small or structural boundaries are less distinct, the performance gap between NGACL and CAGM narrows. These social and email communication networks often exhibit dispersed information propagation chains, unclear community boundaries, and a higher proportion of noisy nodes, limiting the expressiveness of community-based features and consequently reducing the effectiveness of the community-enhanced module in NGACL.

In summary, NGACL demonstrates clear advantages in networks rich in structural information or with well-defined community structures, making it particularly suitable for scenarios with strong local clustering or evident community partitions.

6. Conclusion

This study addresses the problem of identifying node importance in dynamic networks, with the objective of improv-

ing the accuracy of predicting nodes' spreading capability in evolving environments. A novel method, termed NGACL, was developed based on a graph attention mechanism. The approach integrates two types of low-cost structural features, degree centrality and H-index, along with community-level centrality features derived from the Louvain algorithm, enabling the joint modeling of local and global structural characteristics.

Comprehensive experiments conducted on various synthetic and real-world dynamic networks have validated the effectiveness of the NGACL framework. Evaluated using Kendall's τ rank correlation, the method was consistently found to outperform both traditional centrality measures (*e.g.*, degree and betweenness centrality) and learning-based approaches, including LCNN and the communicability-adaptive gravity model (CAGM), demonstrating superior predictive accuracy and computational efficiency across diverse datasets.

The primary contributions of this work can be summarized as follows.

(i) The importance of structural feature design in predictive performance is emphasized. In particular, the integration of local centrality and community-based features was found to significantly enhance prediction accuracy in structurally rich networks.

(ii) The graph attention mechanism was shown to be highly adaptable to varying structural patterns, facilitating robust and effective information extraction in dynamic environments.

(iii) The potential of deep learning for dynamic network modeling was further demonstrated. Despite being based on static temporal snapshots, the proposed method consistently outperformed baseline models, providing useful insights for future research on dynamic graph representations.

In future work, as dynamic networks become increasingly prevalent in domains such as social interaction, communication, and recommendation systems, growing emphasis is expected to be placed on models with stronger temporal sensitivity and generalization capability. Building on the NGACL framework, several promising directions can be pursued: (I) incorporating temporal modeling techniques to more accurately capture network evolution; (II) enhancing adaptability to heterogeneous structural patterns to improve generalization across domains; (III) integrating advanced graph neural architectures and multi-modal data to enrich representational power; and (IV) improving interpretability and computational efficiency to facilitate real-time and transparent deployment.

In summary, NGACL offers a structurally grounded, efficient, and accurate approach for predicting node importance in dynamic networks. The proposed framework lays a strong foundation for related tasks, including information diffusion modeling, influential node detection, and network intervention design, while also contributing to the broader development of theory and applications in dynamic network analysis. From a physics perspective, the NGACL framework can be viewed as analogous to phase transition phenomena, where identifying influential nodes corresponds to locating critical points that govern large-scale structural and dynamical changes in complex networks.

Appendix A: Data availability

The dataset used in this study is publicly available at https://github.com/yumingzz/NGACL GitHub repository.

Appendix B: Glossary

Table B1. Notation and definitions.

Symbol	Description
$\mathcal{T}$	Training set
$\mathcal{T}_e$	Test set
v	Node index
$N(v)$	Neighbor node set of node v
$G_v^{D_1}, G_v^{D_2}, G_v^{D_3}$	Global degree centrality features of node v (1st, 2nd, 3rd order)
$C_v^{D_1}, C_v^{D_2}, C_v^{D_3}$	Degree centrality features of node v within its community (1st, 2nd, 3rd order)
$G_v^{H_1}, G_v^{H_2}, G_v^{H_3}$	Global H-index features of node v (1st, 2nd, 3rd order)
$C_v^{H_1}, C_v^{H_2}, C_v^{H_3}$	H-index features of node v within its community (1st, 2nd, 3rd order)
$D_v^{\text{network}}, D_v^{\text{community}}$	Normalized degree centrality feature vectors (global / community)
$H_v^{\text{network}}, H_v^{\text{community}}$	Normalized H-index feature vectors (global / community)
$\hat{y}$	Predicted importance score by the model
y	Ground truth importance label of node
b	Weight parameter for contrastive loss
$\mathcal{L}_{\text{rank}}$	Pairwise ranking loss
$\mathcal{L}_{\text{con}}$	Contrastive learning loss (InfoNCE)
$\mathcal{L}_{\text{total}}$	Total loss function

References

[1] Battiston F, Cencetti G, Iacopini I, *et al*. 2020 *Phys. Rep.* **874** 1

[2] Fan C, Zeng L, Sun Y, *et al*. 2020 *Nature Machine Intelligence* **2** 317

[3] Guo S, Zhou D, Fan J, *et al*. 2019 *EPJ Data Science* **8** 28

[4] Chakrapani H B, Chourasia S, Gupta S, *et al*. 2021 *Computers in Biology and Medicine* **133** 104378

[5] Fang X and Hu P J H 2018 *MIS Quarterly* **42** 63

[6] C F L 1978 *Social Networks* **1** 215

[7] Chen D, Lü L, Shang M S, *et al*. 2012 *Physica A* **391** 1777
[8] Namtirtha A, Dutta A and Dutta B 2018 *Proc. 10th Int. Conf. Communication Systems and Networks (COMSNETS)* pp. 81–88
[9] Freeman L C 1977 *Sociometry* **40** 35
[10] Sabidussi G 1966 *Psychometrika* **31** 581
[11] Bonacich P 1972 *Journal of Mathematical Sociology* **2** 113
[12] Zhao Z, Wang X, Zhang W and Zhu Z 2015 *Entropy* **17** 2228
[13] Ghalmane Z, El Hassouni M, Cherifi C and Cherifi H 2019 *EPJ Data Science* **8** 15
[14] Asgharian Rezaei A, Munoz J, Jalili M and Khayyam H 2023 *Expert Systems with Applications* **214** 119086
[15] Hu G, Xu X, Zhang W M and Zhou Y 2019 *Acta Electronica Sinica* **47** 358
[16] Yu E Y, Wang Y P, Fu Y, *et al*. 2020 *Knowledge-Based Systems* **198** 105893
[17] Ou Y, Guo Q, Xing J L and Liu J G 2022 *Expert Systems with Applications* **203** 117515
[18] Ahmad W, Wang B and Chen S 2024 *Applied Intelligence* **54** 3260
[19] Blondel V D, Guillaume J L, Lambiotte R and Lefebvre E 2008 *J. Stat. Mech.: Theory Exp.* **2008** P10008
[20] Veličković P, Cucurull G, Casanova A, Romero A, Liò P and Bengio Y 2018 *Proc. Int. Conf. Learning Representations* (ICLR)
[21] Ying R, He R, Chen K, Eksombatchai P, Hamilton W L and Leskovec J 2018 *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining* (KDD) pp. 974–983
[22] Cao H, Wei L, Zhou W and Hu S 2024 arXiv 2407.10474
[23] Guo Y, Tang C, Wu H and Chen B 2024 *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing* (ICASSP) pp. 1921–1925
[24] Gutmann M and Hyvärinen A 2010 *Proc. 13th Int. Conf. Artificial Intelligence and Statistics* (AISTATS) pp. 297–304
[25] van den Oord A, Li Y and Vinyals O 2018 arXiv 1807.03748
[26] Jiang C, *et al.* 2024 *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (CVPR) pp. 27036–27045
[27] Sung C, *et al.* 2024 *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (CVPR) pp. 3732–3741
[28] Hethcote H W 2000 *SIAM Rev.* **42** 599
[29] Ahmad W, Wang B and Chen S 2024 *Appl. Intell.* **54** 3260
[30] Priebe C E, Conroy J M, Marchette D J and Park Y 2005 *Comput. Math. Organ. Theory* **11** 229
[31] Rahman M and Hasan M A 2016 *Mach. Learn. Knowl. Discov. Databases* **9851** 394
[32] Viswanath B, Mislove A, Cha M and Gummadi K P 2009 *Proc. 2nd ACM Workshop on Online Social Networks* pp. 37–42
[33] Mikolov T, Chen K, Corrado G and Dean J 2013 arXiv 1301.3781
[34] Barabási A L and Albert R 1999 *Science* **286** 509
[35] Su X, Cheng J, Yang H, Leng M, Zhang W and Chen X 2020 *Int. J. Mod. Phys. C* **31** 2050094
[36] Greene D, Doyle D and Cunningham P 2010 *Proc. Int. Conf. Advances in Social Networks Analysis and Mining* (ASONAM) pp. 176–183
[37] Wang Z, Zhao Y, Xi J and Du C 2016 *Physica A* **461** 171
[38] Dorogovtsev S N, Goltsev A V and Mendes J F F 2006 *Phys. Rev. Lett.* **96** 040601
[39] Bonacich P 1987 *Am. J. Sociol.* **92** 1170
[40] Xu G and Dong C 2024 *Expert Syst. Appl.* **235** 121154