Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

HGIRank: Multi-View Hypergraph Reasoning for Influential Node Identification in Complex Networks

Lu Xia
Yunnan Normal University
Yongtao Chen
Yunnan Normal University
Yunfei Zi
Yunnan Normal University
Lin Liu
Yunnan Normal University
Long Wang
Kunming Longhui Zhichuang Technology Co., Ltd.
Mingjing Tang
tmj@ynnu.edu.cn
Yunnan Normal University

Research Article

Keywords: Influential node identification, Complex networks, Hypergraph learning, High-order reasoning, Cross-network generalization

Posted Date: September 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11036859/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

HGIRank: Multi-View Hypergraph Reasoning for Influential Node Identification in Complex Networks

Abstract

Influential node identification is fundamental to understanding and controlling diffusion processes in complex networks. Existing topology-based methods rely on predefined structural assumptions, whereas learning-based approaches mainly exploit local neighborhoods, handcrafted structural features, or pairwise graph message passing, making it difficult to capture complementary high-order structural contexts and generalize across heterogeneous networks. To address this limitation, we propose HGIRank(HyperGraph-based Influential Node Ranking), a multi-view hypergraph reasoning framework for influential node identification. HGIRank constructs three complementary hypergraph views from an ordinary graph, namely ego, diffusion, and core views, to represent local neighborhood patterns, truncated short-range expansion, and core-shell structural roles. An adaptive node–hyperedge–hyperedge–node reasoning mechanism then models non-uniform node contributions, hyperedge reliability, overlap-aware interactions, and hyperedge-to-node feedback, followed by role-adaptive fusion of view-specific representations. We evaluate HGIRank under a synthetic-to-real transfer protocol, where learning-based methods are trained on a BA(1000, 4) source network and directly evaluated on 18 unseen real-world networks under 11 infection strengths without target-network labels or fine-tuning. HGIRank achieves the best overall average ranking performance among 14 baseline methods, while maintaining strong ranking consistency and top-ranked node identification across heterogeneous networks. Ablation results further confirm the contributions of multi-view structural representation and adaptive high-order reasoning.

Keywords: Influential node identification, Complex networks, Hypergraph learning, High-order reasoning, Cross-network generalization

1 Introduction

Complex networks provide an effective abstraction for modeling interactions among entities in real-world systems. User interactions in social networks, inter-city connections in transportation networks, protein interactions in biological networks, and co-authorship relations in scientific collaboration networks can all be represented as

graphs composed of nodes and edges (Newman 2003). In these networks, nodes often differ significantly in their structural roles and spreading influence. A small number of influential nodes can play key roles in information diffusion, epidemic spreading, rumor control, traffic congestion mitigation, and network robustness maintenance (Lu et al. 2016). Therefore, accurately identifying influential nodes and generating reliable influence rankings are fundamental to understanding and controlling spreading processes in complex networks (Lu et al. 2016; Chen and Su 2023).

Existing influential node identification methods have made substantial progress, but their representation capabilities remain limited. Classical topology-based methods estimate node importance using predefined structural measures such as degree, betweenness, and coreness. These methods are generally simple and interpretable, but their effectiveness depends on the structural information encoded by the selected centrality measures (Bucur 2020; Kitsak et al. 2010; Lü et al. 2016). Recent learning-based methods improve representation flexibility by learning from neighborhood information, structural features, or graph neural network representations (Yu et al. 2020; Ou et al. 2022; Ahmad et al. 2024; Xiong et al. 2024). However, many existing learning-based approaches still rely primarily on local neighborhoods, node-level structural features, or pairwise message passing (Yu et al. 2020; Ou et al. 2022; Zhao et al. 2020; Wu et al. 2024). Such representations are not well suited to explicitly encode high-order structural units formed by multiple nodes, nor to capture interactions among these units. Hypergraphs and other higher-order representations provide a principled way to model interactions that cannot be fully described by pairwise edges (Battiston et al. 2020, 2021; Bick et al. 2023). However, many hypergraph learning methods are formulated on an explicitly specified hypergraph, where the hyperedge structure is treated as part of the input representation (Feng et al. 2019; Yadati et al. 2019; Chien et al. 2022). In ordinary complex networks, however, the input is typically a pairwise graph rather than an explicitly specified hypergraph. The high-order structural units that may provide useful evidence for node importance therefore need to be derived from the available topology. This creates a representation gap between ordinary pairwise topology and the complementary high-order structural evidence needed for influential node identification.

Beyond structural representation, an important question is whether learned evidence of node influence can transfer across networks with different structural characteristics. The transferability of graph neural networks to unseen graph structures can depend on how learned representations capture graph structure (Thang et al. 2022). For influential node identification, this issue is particularly important because the relationship between structural position and spreading ability depends on both network topology and propagation conditions. Therefore, strong performance under conventional within-network evaluation does not necessarily establish reliable transfer to previously unseen networks. Reliable cross-network transfer should instead be evaluated by training on a source network and directly applying the learned model to unseen target networks without target-network labels or target-specific fine-tuning.

To address these representation and transfer challenges, we propose HGIRank, a multi-view hypergraph reasoning framework for influential node identification in ordinary complex networks. The core idea is to construct multiple complementary

hypergraph views from an ordinary pairwise graph and learn node importance through adaptive reasoning over node–hyperedge memberships and hyperedge–hyperedge interactions. Specifically, HGIRank constructs three hypergraph views from the original graph: an ego view for local neighborhood patterns, a diffusion view for truncated short-range expansion, and a core view for core-shell structural roles. Rather than directly applying a generic hypergraph learning model to a converted graph, HGIRank explicitly constructs informative hyperedges from the original topology to represent complementary high-order structural contexts. It then performs adaptive node–hyperedge–hyperedge–node reasoning to model non-uniform node contributions, estimate hyperedge reliability, propagate information among overlapping hyperedges, and feed hyperedge-level evidence back to nodes. Finally, a role-adaptive multi-view fusion module integrates the complementary representations to produce the final node importance ranking.

To examine cross-network generalization, we evaluate HGIRank under a unified synthetic-to-real transfer protocol. All learning-based methods are trained on the same BA(1000, 4) source network and directly evaluated on 18 unseen real-world networks without target-network labels or target-specific fine-tuning. The target networks are evaluated across 11 infection strengths from $\alpha = 1.0$ to $\alpha = 2.0$. We compare HGIRank with 14 representative baselines and conduct ablation studies to examine the contributions of multi-view hypergraph construction and adaptive high-order reasoning.

The main contributions of this paper are summarized as follows:

- We propose a multi-view hypergraph construction framework for influential node identification in complex networks. By constructing ego, diffusion, and core views from a pairwise graph, HGIRank converts local neighborhood patterns, truncated short-range expansion, and core-shell structural roles into explicit and learnable high-order structural units.
- We develop an adaptive node–hyperedge–hyperedge–node reasoning mechanism for high-order influence representation learning. It models node contributions, hyperedge reliability, and interactions among overlapping hyperedges to capture structural evidence beyond pairwise message passing.
- We design a role-adaptive multi-view fusion module to integrate complementary structural evidence at the node level. Instead of assigning fixed weights to different views, HGIRank learns node-specific view weights, enabling different nodes to rely on different structural contexts when estimating their importance.
- We establish a unified synthetic-to-real transfer protocol for evaluating cross-network generalization in influential node identification. HGIRank and learning-based baselines are trained on the same synthetic source network and directly evaluated on 18 unseen real-world networks under 11 infection strengths without target-network labels or fine-tuning. Comparisons with 14 representative baselines and ablation studies provide a systematic evaluation of the proposed framework.

2 Related work

Influential node identification in complex networks has been widely studied from both topology-based and learning-based perspectives. Classical approaches estimate node importance using predefined structural measures, such as degree centrality (Nieminen 1974), H-index (Lü et al. 2016), K-shell decomposition (Kitsak et al. 2010), PageRank (Page et al. 1999), and VoteRank (Li et al. 2022). These methods are generally efficient and interpretable, but their effectiveness depends on manually designed structural descriptors and may emphasize only particular aspects of network topology. Learning-based approaches improve representation flexibility by learning the relationship between network structure and node importance from data. For example, RCNN (Yu et al. 2020) learns from local neighborhood matrices, M-RCNN (Ou et al. 2022) incorporates multi-level structural attributes, and LCNN (Ahmad et al. 2024) exploits one-hop neighborhoods with multi-scale structural weights. More recent graph neural network approaches, including InfGCN (Zhao et al. 2020), CGNN (Zhang et al. 2022), ReGCN (Wu et al. 2024), AGNN (Xiong et al. 2024), and transformer-based models (Jiang et al. 2025), further learn node representations through graph convolution, structural feature construction, attention, or multi-scale feature fusion. Although these methods improve the flexibility of representation learning, they are still mainly built on ordinary pairwise graphs. Their representations rely primarily on node-level features, local subgraphs, neighborhood matrices, handcrafted structural channels, or pairwise message passing, making it difficult to explicitly model high-order structural contexts jointly formed by multiple nodes.

Hypergraphs provide a natural representation for higher-order relations, where a hyperedge can connect multiple nodes and describe group-level dependencies beyond pairwise edges (Battiston et al. 2020; Bick et al. 2023). Existing studies have extended node-importance estimation to hypergraphs through weighted hypergraph degree (Kapoor et al. 2013), hypercore decomposition (Lee et al. 2023; Mancastroppa et al. 2023), hypergraph eigenvector centrality (Benson 2019), and higher-order centrality (Zhao et al. 2023). Hypergraph neural networks, including HGNN (Feng et al. 2019), HyperGCN (Yadati et al. 2019), and AllSet (Chien et al. 2022), further introduce learnable mechanisms for high-order representation learning. Recent studies have also explored hypergraph-based methods for critical-node analysis and related structural learning tasks, including hypercore-guided convolution (Miao et al. 2024), active hypergraph attention (Ni et al. 2026), critical-node-aware hypergraph contrastive learning (Li et al. 2025), and member-aware attention for hypergraph structural representation (Xie et al. 2025). However, most existing hypergraph-based methods are designed for native hypergraph data, where the hypergraph structure and its hyperedges are assumed to be available before model learning. In ordinary complex networks, the input is typically a pairwise graph, and the high-order structural units useful for node importance modeling are not explicitly given. Therefore, directly applying a generic hypergraph model to a converted graph does not necessarily provide informative and complementary high-order structural contexts for influential node identification.

Beyond representation capability, the generalization of learned graph representations to unseen networks is also an important consideration. Previous studies have

shown that the transferability of graph neural networks can depend on how structural information is represented and on the relationship between the source and unseen target graphs (Thang et al. 2022). This issue is particularly relevant to influential node identification because the relationship between structural position and spreading ability depends on both network topology and propagation conditions. Consequently, strong performance under conventional within-network evaluation does not necessarily demonstrate reliable generalization to previously unseen networks. A stricter evaluation should train a model on a source network and directly apply it to unseen target networks without using target-network labels or target-specific fine-tuning. From this perspective, existing studies leave two closely related challenges insufficiently addressed: how to construct informative high-order structural representations directly from ordinary pairwise networks, and how to evaluate whether the resulting influence representations can transfer to heterogeneous unseen networks without target supervision. These challenges motivate HGIRank, which integrates multi-view hypergraph construction, adaptive high-order reasoning, and role-adaptive fusion within a unified framework for influential node identification and cross-network generalization.

3 Proposed method

Given an undirected and unweighted complex network $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ denotes the node set and E denotes the edge set, influential node identification aims to estimate the importance of nodes and discover those that play dominant roles in the network. In this paper, SIR-induced influence labels are used only for supervision and evaluation, rather than as model inputs. Formally, HGIRank learns a scoring function

$$s_i = f_\theta(v_i, G), \tag{1}$$

where s_i denotes the predicted importance score of node v_i. Nodes with larger scores are considered more influential and are therefore ranked higher in the predicted node importance list.

As illustrated in Fig. 1, HGIRank estimates node importance through a multi-view hypergraph reasoning framework. It first constructs ego, diffusion, and core hypergraph views from the original graph to represent local neighborhood patterns, truncated short-range expansion, and core-shell structural roles. Each view is then converted into a unified hypergraph representation consisting of node features, hyperedge features, node–hyperedge incidence, and sparse hyperedge overlap relations. Based on this representation, HGIRank performs adaptive node–hyperedge–hyperedge–node reasoning to integrate node-level evidence, high-order structural contexts, overlap-aware hyperedge interactions, and feedback from hyperedges to nodes. Finally, role-adaptive multi-view fusion combines the view-specific representations and predicts the final node importance scores.

The following subsections present the main components of HGIRank, including multi-view hypergraph construction, adaptive high-order reasoning, multi-view fusion, and model training.

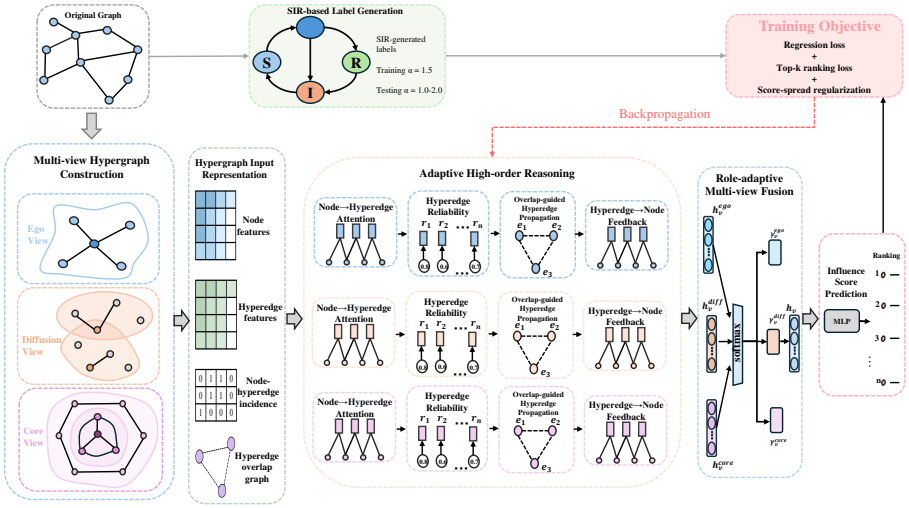


Fig. 1 Overall framework of HGIRank. Starting from an ordinary graph, HGIRank constructs three complementary hypergraph views, namely ego, diffusion, and core views, to capture local neighborhood patterns, truncated short-range expansion, and core-shell structural roles. On the constructed views, HGIRank performs adaptive node–hyperedge–hyperedge–node reasoning to learn non-uniform node contributions, hyperedge reliability, overlap-aware high-order interactions, and hyperedge-to-node feedback. The resulting view-specific node representations are adaptively fused for influence score prediction and ranking, while SIR-generated labels are used only for training supervision.

3.1 Multi-view hypergraph construction

The key idea of HGIRank is not to directly apply a generic hypergraph model to an ordinary graph, but to construct informative hyperedges from the original graph so that different high-order structural contexts can be explicitly represented. In ordinary complex networks, node importance is usually shaped by multiple complementary structural factors, including local neighborhood patterns, short-range expansion ability, and core-level structural position. Accordingly, HGIRank constructs three hypergraph views from the original graph: the ego view, the diffusion view, and the core view.

For each view, the constructed hypergraph is denoted as

$$\mathcal{H}^{(r)} = (V, \mathcal{E}^{(r)}), \tag{2}$$

where $r \in \{\text{ego}, \text{diff}, \text{core}\}$, V is the original node set, and $\mathcal{E}^{(r)}$ is the hyperedge set under view r. Since directly collecting all nodes in a neighborhood or a shell may produce overly large hyperedges, especially around hub nodes or dense core layers, HGIRank keeps hyperedges compact through candidate truncation. When truncation is required, a lightweight structural priority score $\pi(v)$ is used to rank candidate nodes. This score combines several low-cost structural indicators, including degree, core number, H-index, two-hop reachability, and PageRank. It is used only for candidate selection and hyperedge size control, rather than as an influence label or prediction target. Duplicate hyperedges are removed after each view is constructed.

3.1.1 Ego view

The ego view captures the direct local context of a node. For node v_i, let $\mathcal{N}_i^{\text{ego}}$ denote the retained one-hop neighbors of v_i. The ego hyperedge is defined as

$$e_i^{\text{ego}} = \{v_i\} \cup \mathcal{N}_i^{\text{ego}}. \tag{3}$$

If the number of one-hop neighbors exceeds the maximum hyperedge size, only the top-ranked neighbors according to $\pi(\cdot)$ are retained in $\mathcal{N}_i^{\text{ego}}$. This view extends local pairwise contacts around v_i into a high-order neighborhood unit, enabling the subsequent reasoning module to model local structural evidence at the group level.

3.1.2 Diffusion view

The diffusion view captures compact short-range expansion beyond immediate neighbors. The ego view only describes direct neighbors, while the structural importance of a node may also depend on its ability to reach a broader short-range region through intermediate nodes. However, directly using all two-hop neighbors may introduce noisy candidates and produce oversized hyperedges. Therefore, HGIRank constructs a truncated two-hop expansion view rather than a complete two-hop closure.

For node v_i, HGIRank first selects a bounded set of structurally informative one-hop neighbors, denoted as $\mathcal{N}_1^T(v_i)$. It then expands from these selected one-hop neighbors and retains structurally important two-hop candidates, denoted as $\mathcal{N}_2^*(v_i)$. Let $\mathcal{N}_i^{\text{diff}}$ denote the finally retained short-range candidates. The diffusion construction can be written as

$$\begin{aligned} \mathcal{N}_i^{\text{diff}} &\subseteq \mathcal{N}_1^T(v_i) \cup \mathcal{N}_2^*(v_i), \\ e_i^{\text{diff}} &= \{v_i\} \cup \mathcal{N}_i^{\text{diff}}. \end{aligned} \tag{4}$$

In this construction, one-hop and two-hop candidates are not treated equally. One-hop candidates provide stronger local evidence, while two-hop candidates are considered through selected intermediate neighbors and further filtered by structural priority. A final top-ranked truncation is applied to keep the diffusion hyperedge compact. Thus, the diffusion view preserves useful short-range reachability information without degenerating into an unbounded two-hop neighborhood.

3.1.3 Core view

Besides local neighborhoods and short-range expansion, node importance is also related to its position in the core–periphery structure. Nodes located in higher core layers are usually embedded in denser regions and may have stronger structural support. However, directly using the core number as a scalar feature is often too coarse, because many nodes may share the same shell value. Therefore, HGIRank constructs the core view from both shell-level hierarchy and local high-core connectivity.

For each core level q, the shell set is defined as

$$S_q = \{v_i \in V \mid \kappa(v_i) = q\}, \tag{5}$$

where $\kappa(v_i)$ denotes the core number of node v_i. Nodes in the same shell are grouped into shell-level hyperedges. If a shell contains too many nodes, it is partitioned into smaller hyperedges according to the structural priority score, so that each hyperedge remains compact and avoids overly coarse grouping.

In addition to shell-level hyperedges, HGIRank constructs local high-core hyperedges for selected high-core nodes. For a high-core node v_i, let $\mathcal{N}_i^{\mathrm{hc}}$ denote the retained high-core neighbors satisfying the local core constraint. Specifically, a neighbor $u \in \mathcal{N}(v_i)$ can be included in $\mathcal{N}_i^{\mathrm{hc}}$ only if $\kappa(u) \geq \tau_i$, where $\tau_i = \max\{2, \kappa(v_i) - 1\}$. The local high-core hyperedge is defined as

$$e_i^{\mathrm{core}} = \{v_i\} \cup \mathcal{N}_i^{\mathrm{hc}}. \tag{6}$$

If the local high-core candidate set exceeds the maximum hyperedge size, the top-ranked nodes according to $\pi(\cdot)$ are retained. In this way, the core view preserves both global core-shell hierarchy and local high-core structural relations, which are difficult to capture by the ego and diffusion views alone.

By constructing the above three views, HGIRank transforms an ordinary pairwise graph into complementary high-order structural contexts. The ego view focuses on direct local neighborhoods, the diffusion view captures truncated short-range expansion, and the core view describes core-shell hierarchy together with local high-core connectivity. These complementary views provide the structural foundation for the subsequent adaptive node–hyperedge–hyperedge–node reasoning module.

3.2 Hypergraph input representation

After the hypergraph views are constructed, each view is converted into a unified representation so that the same reasoning module can be applied to all views. For view r, let $\mathcal{E}^{(r)} = \{e_1^{(r)}, e_2^{(r)}, \ldots, e_{L_r}^{(r)}\}$ denote its hyperedge set. HGIRank represents this view as

$$\mathcal{I}^{(r)} = \left(\mathbf{X}^{(r)}, \mathbf{U}^{(r)}, \mathbf{B}^{(r)}, \mathcal{R}^{(r)}\right), \tag{7}$$

where $\mathbf{X}^{(r)} \in \mathbb{R}^{n \times d_v}$ and $\mathbf{U}^{(r)} \in \mathbb{R}^{L_r \times d_e}$ denote the node feature matrix and the hyperedge feature matrix, respectively. These features are derived from hypergraph participation and overlap statistics, such as incidence degree, hyperedge size, member-node statistics, coverage, and redundancy-related information.

The node–hyperedge membership relation is encoded by the incidence matrix

$$\mathbf{B}_{im}^{(r)} = \begin{cases} 1, & v_i \in e_m^{(r)}, \\ 0, & v_i \notin e_m^{(r)}. \end{cases} \tag{8}$$

This matrix determines which nodes can send messages to a hyperedge and which hyperedges can feed information back to a node.

To further model interactions among high-order structural units, HGIRank constructs a sparse hyperedge overlap relation. For two hyperedges $e_p^{(r)}$ and $e_q^{(r)}$, their

overlap strength is measured by

$$w_{pq}^{(r)} = \frac{|e_p^{(r)} \cap e_q^{(r)}|}{|e_p^{(r)} \cup e_q^{(r)}|}. \tag{9}$$

Only retained pairs with nonzero overlap are stored in $\mathcal{R}^{(r)}$. This sparse representation avoids dense all-to-all hyperedge interactions while preserving useful overlap information. Based on $\mathcal{I}^{(r)}$, HGIRank performs adaptive high-order reasoning in each view.

3.3 Adaptive high-order reasoning over node–hyperedge interactions

Given the hypergraph input representation $\mathcal{I}^{(r)} = (\mathbf{X}^{(r)}, \mathbf{U}^{(r)}, \mathbf{B}^{(r)}, \mathcal{R}^{(r)})$ for view $r \in \{\text{ego}, \text{diff}, \text{core}\}$, HGIRank performs adaptive high-order reasoning to learn node-importance representations. The constructed hyperedges provide candidate high-order structural units, but they do not directly indicate the contribution of each member node, the reliability of each unit, or the interactions among overlapping units. Therefore, HGIRank models the reasoning process along the path from nodes to hyperedges, from hyperedges to overlapping hyperedges, and finally from hyperedges back to nodes.

For simplicity, we describe the reasoning process in one view and omit the superscript r when no ambiguity arises. In the following, $\mathbf{x}_i$, $\mathbf{u}_m$, $\mathbf{B}$, and $\mathcal{R}$ refer to the node feature, hyperedge feature, incidence matrix, and hyperedge-overlap relation of the current view, respectively. Let e_m denote the m-th hyperedge in the current view. According to the incidence matrix $\mathbf{B}$, the member nodes of e_m and the incident hyperedges of node v_i are defined as

$$\mathcal{V}_m = \{v_i \mid \mathbf{B}_{im} = 1\}, \quad \mathcal{E}_i = \{e_m \mid \mathbf{B}_{im} = 1\}. \tag{10}$$

The initial node and hyperedge representations are obtained by

$$\mathbf{h}_i^{(0)} = \phi_v(\mathbf{x}_i), \quad \mathbf{g}_m^{(0)} = \phi_e(\mathbf{u}_m), \tag{11}$$

where $\phi_v(\cdot)$ and $\phi_e(\cdot)$ are learnable projection functions. Here, $\sigma(\cdot)$ denotes a nonlinear activation function, and all transformation matrices, attention vectors, and bias terms are learnable parameters with compatible dimensions.

3.3.1 Attentive node-to-hyperedge aggregation

A hyperedge represents a high-order structural unit composed of multiple nodes. However, different member nodes may contribute unequally to the unit representation. For node $v_i \in \mathcal{V}_m$, HGIRank computes the node-to-hyperedge attention score as

$$a_{im}^{(l)} = \text{LeakyReLU}\left(\mathbf{a}_n^\top [\mathbf{W}_h \mathbf{h}_i^{(l)} \| \mathbf{W}_g \mathbf{g}_m^{(l)}]\right). \tag{12}$$

The normalized contribution of node v_i to hyperedge e_m is

$$\alpha_{im}^{(l)} = \frac{\exp(a_{im}^{(l)})}{\sum_{v_j \in \mathcal{V}_m} \exp(a_{jm}^{(l)})}. \tag{13}$$

Then, the node-to-hyperedge message and the updated hyperedge representation are computed as

$$\begin{aligned} \mathbf{m}_m^{(l)} &= \sum_{v_i \in \mathcal{V}_m} \alpha_{im}^{(l)} \mathbf{W}_h \mathbf{h}_i^{(l)}, \\ \widehat{\mathbf{g}}_m^{(l)} &= \sigma\left(\mathbf{W}_s \mathbf{g}_m^{(l)} + \mathbf{m}_m^{(l)}\right). \end{aligned} \tag{14}$$

This step aggregates node-level information into hyperedge-level representations while preserving non-uniform node contributions.

3.3.2 Reliability-aware hyperedge calibration

The constructed hyperedges are not equally informative. Some hyperedges may contain noisy or redundant members, especially in dense neighborhoods or large structural contexts. To reduce the influence of low-quality units, HGIRank estimates a reliability score for each hyperedge:

$$\lambda_m^{(l)} = \sigma\left(\mathbf{w}_\lambda^\top \widehat{\mathbf{g}}_m^{(l)} + b_\lambda\right), \tag{15}$$

where $\lambda_m^{(l)} \in (0, 1)$. The reliability score is learned without additional supervision and is used to modulate subsequent hyperedge messages.

3.3.3 Overlap-guided hyperedge propagation

High-order structural units are not isolated. Two hyperedges may share nodes and thus correspond to overlapping structural contexts. Based on the sparse hyperedge-overlap relation $\mathbf{R}$, HGIRank propagates information among overlapping hyperedges. The overlap neighbor set of hyperedge e_m is defined as

$$\mathcal{N}_m = \{e_p \mid (p, m, w_{pm}) \in \mathbf{R}\}, \tag{16}$$

where w_{pm} denotes the overlap strength between hyperedges e_p and e_m. The reliability-aware propagation coefficient from e_p to e_m is then computed as

$$\eta_{pm}^{(l)} = \frac{w_{pm} \lambda_p^{(l)}}{\sum_{e_q \in \mathcal{N}_m} w_{qm} \lambda_q^{(l)} + \epsilon}, \quad e_p \in \mathcal{N}_m, \tag{17}$$

where ϵ is a small constant for numerical stability. The overlap-guided hyperedge message and propagated hyperedge representation are computed as

$$\begin{aligned} \mathbf{m}_{m,\mathrm{ov}}^{(l)} &= \sum_{e_p \in \mathcal{N}_m} \eta_{pm}^{(l)} \mathbf{W}_o \widehat{\mathbf{g}}_p^{(l)}, \\ \widetilde{\mathbf{g}}_m^{(l)} &= \sigma\left(\mathbf{W}_o \widehat{\mathbf{g}}_m^{(l)} + \mathbf{m}_{m,\mathrm{ov}}^{(l)}\right). \end{aligned} \tag{18}$$

This step enables information exchange among overlapping high-order units while avoiding dense all-to-all hyperedge interactions.

3.3.4 Attentive hyperedge-to-node feedback

After hyperedge-level propagation, the refined hyperedge information is fed back to nodes. Since a node may participate in multiple hyperedges, different hyperedges may provide different structural evidence. For each incident hyperedge $e_m \in \mathcal{E}_i$, HGIRank computes the hyperedge-to-node attention score as

$$c_{mi}^{(l)} = \mathrm{LeakyReLU}\left(\mathbf{a}_h^\top [\mathbf{W}_v \mathbf{h}_i^{(l)} \| \mathbf{W}_g \widetilde{\mathbf{g}}_m^{(l)}]\right). \tag{19}$$

The normalized feedback weight from hyperedge e_m to node v_i is

$$\beta_{mi}^{(l)} = \frac{\exp(c_{mi}^{(l)})}{\sum_{e_q \in \mathcal{E}_i} \exp(c_{qi}^{(l)})}. \tag{20}$$

The hyperedge-to-node message and updated node representation are given by

$$\begin{aligned} \mathbf{m}_i^{(l)} &= \sum_{e_m \in \mathcal{E}_i} \beta_{mi}^{(l)} \lambda_m^{(l)} \mathbf{W}_g \widetilde{\mathbf{g}}_m^{(l)}, \\ \mathbf{h}_i^{(l+1)} &= \sigma\left(\mathbf{W}_{\mathrm{res}} \mathbf{h}_i^{(l)} + \mathbf{m}_i^{(l)}\right). \end{aligned} \tag{21}$$

Through this feedback step, each node representation incorporates the high-order structural units it participates in, as well as the interactions among those units.

After stacking L reasoning layers, HGIRank obtains the view-specific node representation

$$\mathbf{z}_i^{(r)} = \mathbf{h}_i^{(L,r)}. \tag{22}$$

The representations from different views are then passed to the multi-view fusion module.

3.4 Role-adaptive multi-view fusion

After adaptive high-order reasoning, each node obtains three view-specific representations, denoted as $\mathbf{z}_i^{(r)}$ with $r \in \{\mathrm{ego}, \mathrm{diff}, \mathrm{core}\}$. These representations describe complementary structural semantics: the ego view emphasizes direct local context, the diffusion view captures compact short-range expansion, and the core view reflects core-level structural roles. Since different nodes may rely on different structural evidence, simply averaging all view representations may introduce irrelevant or noisy signals.

To adaptively fuse the three views, HGIRank learns node-specific view weights. Let $\mathbf{v}_r$ denote a learnable embedding of view r. For node v_i under view r, the view-level attention score is computed as

$$q_i^{(r)} = \mathbf{w}_f^\top \sigma\left(\mathbf{W}_f [\mathbf{z}_i^{(r)} \| \mathbf{v}_r] + \mathbf{b}_f\right), \tag{23}$$

where $\mathbf{W}_f$, $\mathbf{w}_f$, $\mathbf{b}_f$, and $\mathbf{v}_r$ are learnable parameters. The normalized importance of view r for node v_i is then obtained by

$$\gamma_i^{(r)} = \frac{\exp(q_i^{(r)})}{\sum_{r' \in \{\text{ego,diff,core}\}} \exp(q_i^{(r')})}. \tag{24}$$

The final node representation is computed as the weighted combination of the view-specific representations:

$$\mathbf{z}_i = \sum_{r \in \{\text{ego,diff,core}\}} \gamma_i^{(r)} \mathbf{z}_i^{(r)}. \tag{25}$$

The fused representation $\mathbf{z}_i$ is then passed to the prediction layer to estimate the final importance score of node v_i.

3.5 SIR-based label generation and training objective

After role-adaptive multi-view fusion, HGIRank obtains the final representation $\mathbf{z}_i$ for each node v_i. The importance score is predicted by

$$s_i = \text{MLP}_s(\mathbf{z}_i), \tag{26}$$

where a larger score indicates higher predicted node importance. The score s_i is used only for ranking nodes after the graph representation has been learned. Thus, HGIRank separates representation learning from label generation: the model predicts node importance from structural representations, whereas propagation simulations are only used to provide supervision and evaluation references.

The SIR model is adopted to generate node-level supervision labels, rather than as part of the model input. Following common practice in influential node identification, each node is selected as the initial infected node, and its empirical influence is estimated by repeated SIR simulations (Kitsak et al. 2010; Pastor-Satorras et al. 2015). At each time step, an infected node infects each susceptible neighbor with probability β and recovers with probability μ. The process stops when no infected node remains. Fig. 2 illustrates a simple example of the SIR process used for label generation. In this way, SIR simulations provide a graph-dependent empirical ranking signal, while the proposed model itself still relies only on the original network topology and the constructed hypergraph views.

For node v_i, the average final outbreak scale over M simulations is used as its influence label:

$$y_i = \frac{1}{M} \sum_{m=1}^{M} \frac{R_i^{(m)}}{|V|}, \tag{27}$$

where $R_i^{(m)}$ denotes the number of recovered nodes in the m-th simulation starting from v_i. The normalization by $|V|$ makes the label comparable across networks with different sizes and constrains the empirical influence value to a consistent scale. To make infection probabilities comparable across networks, the infection probability is

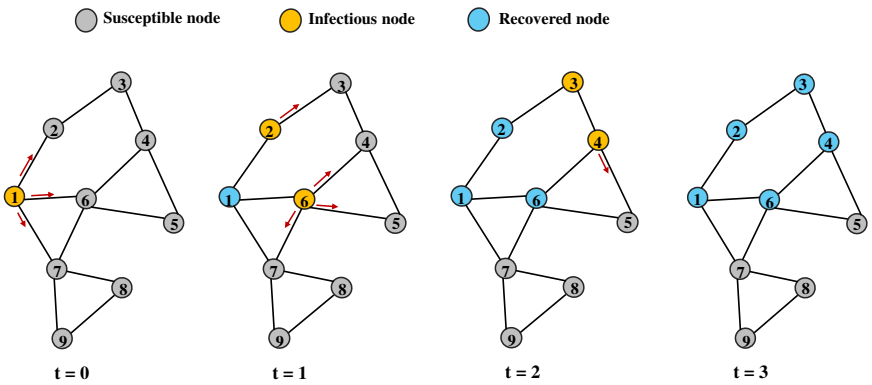


Fig. 2 Illustration of the SIR spreading process used for label generation. Gray, yellow, and blue nodes denote susceptible, infectious, and recovered nodes, respectively. The SIR process is used only to generate supervision labels and evaluation rankings, not as part of the HGIRank input.

set according to the epidemic threshold:

$$\beta_c = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}, \quad \beta = \alpha \beta_c, \tag{28}$$

where α controls the infection strength. The concrete settings of M, μ, and α are provided in the experimental setup. In our source-to-target protocol, these labels are generated on the source network for model training, while target-network labels are used only for evaluation. This setting avoids using target supervision during testing and allows us to assess whether the learned high-order structural patterns can generalize to unseen real-world networks.

HGIRank is optimized by a joint objective that combines score regression, pairwise ranking, and score-dispersion regularization:

$$\mathcal{L} = \lambda_{\text{mse}} \mathcal{L}_{\text{mse}} + \lambda_{\text{rank}} \mathcal{L}_{\text{rank}} + \lambda_{\text{sp}} \mathcal{L}_{\text{sp}}, \tag{29}$$

where $\mathcal{L}_{\text{mse}}$ calibrates the predicted scores with SIR-generated labels, $\mathcal{L}_{\text{rank}}$ encourages pairwise ranking consistency following the learning-to-rank paradigm (Burges et al. 2005), and $\mathcal{L}_{\text{sp}}$ prevents excessive concentration of predicted scores. The balancing coefficients λ_{mse}, λ_{rank}, and λ_{sp} are reported in the experimental setup.

The three terms play complementary roles. The regression term provides pointwise supervision so that the predicted score of each node is aligned with its empirical spreading ability. The pairwise ranking term emphasizes the relative order between nodes, which is important because influential node identification is fundamentally a ranking problem rather than a pure score-fitting task. The score-dispersion term discourages degenerate solutions in which different nodes receive overly similar scores, thereby improving the discriminability of the predicted rankings. By combining these objectives, HGIRank is encouraged to preserve both absolute influence calibration and relative ranking consistency.

After training, HGIRank can be directly applied to an unseen target network without using any target-network labels. Specifically, for a target network, the ego, diffusion, and core hypergraph views are first constructed using the same view construction strategy. The trained high-order reasoning module then produces view-specific node representations, and the role-adaptive fusion module assigns node-dependent weights to different views. Finally, nodes are ranked according to their predicted scores. This inference process is label-free on the target network and is therefore consistent with the transfer setting used in the experiments. Algorithm 1 summarizes the inference procedure of the trained HGIRank model on a target network.

Algorithm 1 Identifying influential nodes through the trained HGIRank model

Require: Input network $G = (V, E)$, trained HGIRank model Θ, view set $\mathcal{R} = \{\text{ego}, \text{diff}, \text{core}\}$, number of reasoning layers L
Ensure: Ranked node list $\mathcal{R}_{\text{rank}}$
Compute the structural priority score $\pi(v)$ for candidate truncation
for each view $r \in \mathcal{R}$ **do**
    Construct the view-specific hypergraph $\mathcal{H}^{(r)} = (V, \mathcal{E}^{(r)})$ from G
    Build the input representation $\mathcal{I}^{(r)} = (\mathbf{X}^{(r)}, \mathbf{U}^{(r)}, \mathbf{B}^{(r)}, \mathcal{R}^{(r)})$
end for
for each view $r \in \mathcal{R}$ **do**
    Initialize node and hyperedge representations from $\mathbf{X}^{(r)}$ and $\mathbf{U}^{(r)}$
    for $l = 1$ to L **do**
        Perform attentive node-to-hyperedge aggregation
        Estimate hyperedge reliability scores
        Propagate information among overlapping hyperedges
        Feed high-order hyperedge information back to nodes
    end for
    Obtain view-specific node representations $\{\mathbf{z}_i^{(r)}\}_{v_i \in V}$
end for
Compute role-adaptive view weights $\gamma_i^{(r)}$ and fused representations $\mathbf{z}_i$
Predict node importance scores $s_i = \text{MLP}_s(\mathbf{z}_i)$ for all $v_i \in V$
Sort nodes in descending order of s_i to obtain $\mathcal{R}_{\text{rank}}$
return $\mathcal{R}_{\text{rank}}$

4 Experimental setup

4.1 Datasets

We use one synthetic network as the training source and 18 real-world networks as target networks for testing. The training network is generated by the Barabási–Albert (BA) model (Barabási and Albert 1999). In this paper, BA$(1000, 4)$ denotes a BA network with 1000 nodes and an average degree close to 4. This source network provides a moderate-sized scale-free topology with heterogeneous node degrees, which is useful for learning node-importance-related structural patterns while keeping SIR label generation and model training computationally manageable.

The target networks are collected from multiple real-world domains, including collaboration, social, trust, communication, transportation, and biological systems. They vary in network scale, degree distribution, clustering coefficient, and connectivity, providing a diverse benchmark for evaluating cross-network generalization. All networks are treated as undirected and unweighted graphs. Table 1 summarizes the

main structural statistics, where n, m, $\langle k \rangle$, $k_{\max}$, c, and LCC denote the number of nodes, the number of edges, the average degree, the maximum degree, the average clustering coefficient, and the proportion of nodes in the largest connected component, respectively.

Table 1 Basic topological features of the source and target networks.

Networks	n	m	$\langle k \rangle$	$k_{\max}$	c	LCC
BA(1000, 4) (Barabási and Albert 1999)	1000	1996	3.992	59	0.027	100.00
Vidal (Rual et al. 2005)	3023	6149	4.068	129	0.066	92.06
Oz (Freeman et al. 1998)	217	1839	16.949	56	0.363	100.00
Hamster (Kunegis 2013)	2426	16631	13.711	273	0.538	82.44
FilmTrust (Guo et al. 2013)	874	1309	2.995	67	0.162	69.79
Faa (Kunegis 2016)	1226	2408	3.928	34	0.068	100.00
USAir (Kunegis 2013)	1574	17215	21.874	314	0.504	99.87
Jazz (Gleiser and Danon 2003)	198	2742	27.697	100	0.617	100.00
PCBI (Rocha et al. 2011)	16730	39044	4.668	305	0.000	94.50
Stelzl (Stelzl et al. 2005)	1702	3155	3.707	95	0.006	94.89
Bible (Kunegis 2013)	1773	9131	10.300	364	0.721	96.28
Figeys (Ewing et al. 2007)	2239	6432	5.745	314	0.040	99.02
LastFM (Rozemberczki and Sarkar 2020)	7624	27806	7.294	216	0.219	100.00
Sex (Rocha et al. 2011)	10106	39016	7.721	311	0.007	100.00
Email (Guimerà et al. 2003)	1133	5451	9.622	71	0.220	100.00
Hep (Leskovec et al. 2007)	9875	25973	5.260	65	0.472	87.47
Arenas (Duch and Arenas 2005)	453	2025	8.940	237	0.646	100.00
NS (Newman 2006)	1461	2742	3.754	34	0.694	25.94
Twitter (McAuley and Leskovec 2012)	23370	32831	2.810	238	0.069	95.52

4.2 Compared methods

We compare HGIRank with 14 representative baselines from three categories. The first category contains traditional topology-based methods, including Degree (Nieminen 1974), K-shell (Kitsak et al. 2010), H-index (Lü et al. 2016), and Betweenness (Freeman 1977). The second category contains hypergraph structural methods, including HyperDegree (Kapoor et al. 2013) and HyperEigen (Benson 2019). The third category contains learning-based methods, including RCNN (Yu et al. 2020), M-RCNN (Ou et al. 2022), LCNN (Ahmad et al. 2024), ReGCN (Wu et al. 2024), InfGCN (Zhao et al. 2020), CGNN (Zhang et al. 2022), MSF-Former (Jiang et al. 2025), and AGNN (Xiong et al. 2024).

For a fair comparison, all learning-based methods follow the same synthetic-to-real transfer protocol whenever applicable. They are trained on the same synthetic source network with the same training labels and then directly evaluated on the real-world target networks. Traditional topology-based methods and hypergraph structural methods do not require training labels and are directly evaluated on the target networks. All methods are compared on the same target networks, infection-strength settings, and evaluation metrics.

4.3 Evaluation metrics

We evaluate node ranking quality from two perspectives: global ranking consistency and top-ranked node identification. For global ranking consistency, Kendall's τ (Knight 1966) and Spearman's ρ (Spearman 1904) are used to measure the correlation between the predicted ranking and the SIR-induced ranking. For top-ranked node identification, we use Jaccard@10% (Ivchenko and Honov 1998), MAP@10% (Manning et al. 2008), and NDCG@10% (Järvelin and Kekäläinen 2002). Specifically, for each network, the cutoff is set to $K = \lceil 0.1n \rceil$, where n is the number of nodes. Jaccard@10% measures the overlap between the predicted and ground-truth top-K node sets. MAP@10% evaluates whether ground-truth top-ranked nodes appear early in the predicted ranking. NDCG@10% further considers ranking positions and uses SIR-generated influence labels as graded relevance scores. For these five raw metrics, larger values indicate better performance.

Since the experiments involve multiple target networks and infection strengths, we further report average-rank indicators to summarize overall performance. For each raw metric, all methods are ranked from best to worst on each network and infection strength, and the average rank is then computed. Let r_τ, r_ρ, r_J, r_M, and r_N denote the average ranks of a method under Kendall's τ, Spearman's ρ, Jaccard@10%, MAP@10%, and NDCG@10%, respectively. We define

$$\text{Tau/Rho Rank} = \frac{r_\tau + r_\rho}{2}, \tag{30}$$

$$\text{Top-}k\text{ Rank} = \frac{r_J + r_M + r_N}{3}, \tag{31}$$

and

$$\text{AvgRank} = \frac{r_\tau + r_\rho + r_J + r_M + r_N}{5}. \tag{32}$$

Tau/Rho Rank summarizes global ranking consistency, Top-k Rank summarizes top-ranked node identification, and AvgRank reflects overall performance across all five raw metrics. Different from the raw metrics, lower values of these rank-based indicators indicate better performance.

4.4 Experimental protocol and implementation settings

We adopt a synthetic-to-real transfer protocol to evaluate the cross-network generalization ability of HGIRank. Under this protocol, all learning-based methods are trained on the same synthetic source network and then directly evaluated on 18 real-world target networks. No labels from the target networks are used for training or fine-tuning, which provides a strict setting for examining whether a method can generalize from synthetic structures to heterogeneous real-world networks.

SIR simulations are used to generate node-level influence labels for both the source and target networks. For the source network, training labels are generated at infection strength $\alpha = 1.5$. For each target network, evaluation labels are generated under 11 infection strengths from $\alpha = 1.0$ to $\alpha = 2.0$ with a step size of 0.1. All methods are evaluated using the same target networks, infection-strength settings, and ranking metrics, ensuring a fair and consistent comparison.

For SIR label generation, each node is selected as the initial infected node and simulated 100 times. The recovery probability is set to $\mu = 1.0$, and the infection probability is computed as $\beta = \alpha\beta_c$, where β_c denotes the epidemic threshold. If the computed infection probability exceeds 1, it is clipped to 1. HGIRank is optimized using Adam (Kingma and Ba 2015). For learning-based baselines, we retain their original model architectures and use the hyperparameter settings reported in the corresponding papers whenever applicable, while retraining all learning-based methods under the same synthetic-to-real transfer protocol and source network. The main implementation settings of HGIRank are summarized in Table 2.

Table 2 Implementation settings of HGIRank.

Item	Setting
Training network	BA(1000, 4)
Training infection strength	$\alpha = 1.5$
Testing infection strengths	$\alpha = 1.0$ to 2.0 with step 0.1
SIR simulations per node	100
Recovery probability	$\mu = 1.0$
Optimizer	Adam
Learning rate	1×10^{-3}
Weight decay	5×10^{-4}
Model configuration	hidden 128, layers 2, dropout 0.20
Training schedule	180 epochs, 20 warmup epochs
Random seeds	1, 2, 3
Loss weights	0.30/1.00/0.01 for MSE/rank/spread

5 Experimental results

5.1 Overall transfer performance across infection strengths

We first evaluate the overall performance of all methods under the synthetic-to-real transfer setting. The comparison is conducted over 18 real-world networks and 11 infection strengths, resulting in 198 evaluation cases for each method. Since the five raw metrics describe different aspects of ranking quality, we report three average-rank indicators for a comprehensive comparison: AvgRank, Tau/Rho Rank, and Top-k Rank. Lower values indicate better performance.

As shown in Table 3, HGIRank achieves the lowest values on all three summary indicators. It obtains an AvgRank of 3.4515, which is 0.6263 lower than that of the second-best method, HyperDegree. This result indicates that HGIRank provides the best overall ranking performance when different networks, infection strengths, and evaluation metrics are jointly considered.

For global ranking consistency, HGIRank obtains a Tau/Rho Rank of 3.5556, slightly better than HyperDegree with 3.5859. For top-ranked node identification, HGIRank achieves a Top-k Rank of 3.3822, which is 1.0235 lower than that of HyperDegree. The larger gain on Top-k Rank suggests that the proposed model is especially effective in identifying highly ranked influential nodes, while still maintaining competitive global ranking consistency.

Table 3 Overall average-rank comparison across transfer tests.

Method	AvgRank↓	T/R Rank↓	Top-k Rank↓
HGIRank	**3.4515**	**3.5556**	**3.3822**
HyperDegree	4.0778	3.5859	4.4057
H-index	5.5495	6.4773	4.9310
HyperEigen	5.9874	4.3081	7.1069
RCNN	6.1010	7.0177	5.4899
K-shell	6.8566	6.7399	6.9343
MSF-Former	7.5227	5.1768	9.0867
Degree	7.8480	8.1086	7.6742
ReGCN	8.2813	7.5783	8.7500
InfGCN	9.3444	10.4242	8.6246
LCNN	9.4949	9.7601	9.3182
CGNN	10.1187	9.4722	10.5497
M-RCNN	10.2000	11.8384	9.1077
Betweenness	12.2783	12.9798	11.8106
AGNN	12.8879	12.9773	12.8283

Among the baselines, HyperDegree achieves the second-best overall performance, showing that high-order structural information is useful for node importance ranking. However, its fixed scoring rule cannot adaptively distinguish informative high-order structural units from noisy or redundant ones. Traditional topology-based methods, such as H-index, K-shell, and Degree, perform competitively on some indicators but are limited by predefined structural assumptions. Learning-based baselines can learn nonlinear structure–importance mappings, but their performance is less stable under the cross-network transfer setting, partly because they rely on local structural matrices, handcrafted feature channels, or pairwise graph message passing.

Overall, the results show that multi-view high-order structural modeling and adaptive high-order reasoning help HGIRank learn more transferable node representations. HGIRank improves the overall average ranking performance and achieves strong results in both global ranking consistency and top-ranked node identification.

5.2 Network-wise global ranking consistency

The overall average-rank results provide a comprehensive comparison across networks, infection strengths, and evaluation metrics, but they do not show how each method behaves on individual networks. Therefore, we further examine network-wise global ranking consistency using Kendall's τ, which measures the pairwise order agreement between the predicted ranking and the SIR-induced ranking.

As shown in Fig. 3, the results are reported on 18 real-world networks under infection strengths from $\alpha = 1.0$ to $\alpha = 2.0$. For visual clarity, Fig. 3 presents HGIRank together with 9 representative baselines, while the complete comparison involving all 14 baselines is reported in Table 3. Each subfigure corresponds to one target network, and each curve shows how the performance of a method changes with the infection strength. The curves show that HGIRank maintains competitive Kendall's τ values across most networks and infection strengths. Although it is not always the best method on every individual network, it remains among the leading methods in a wide

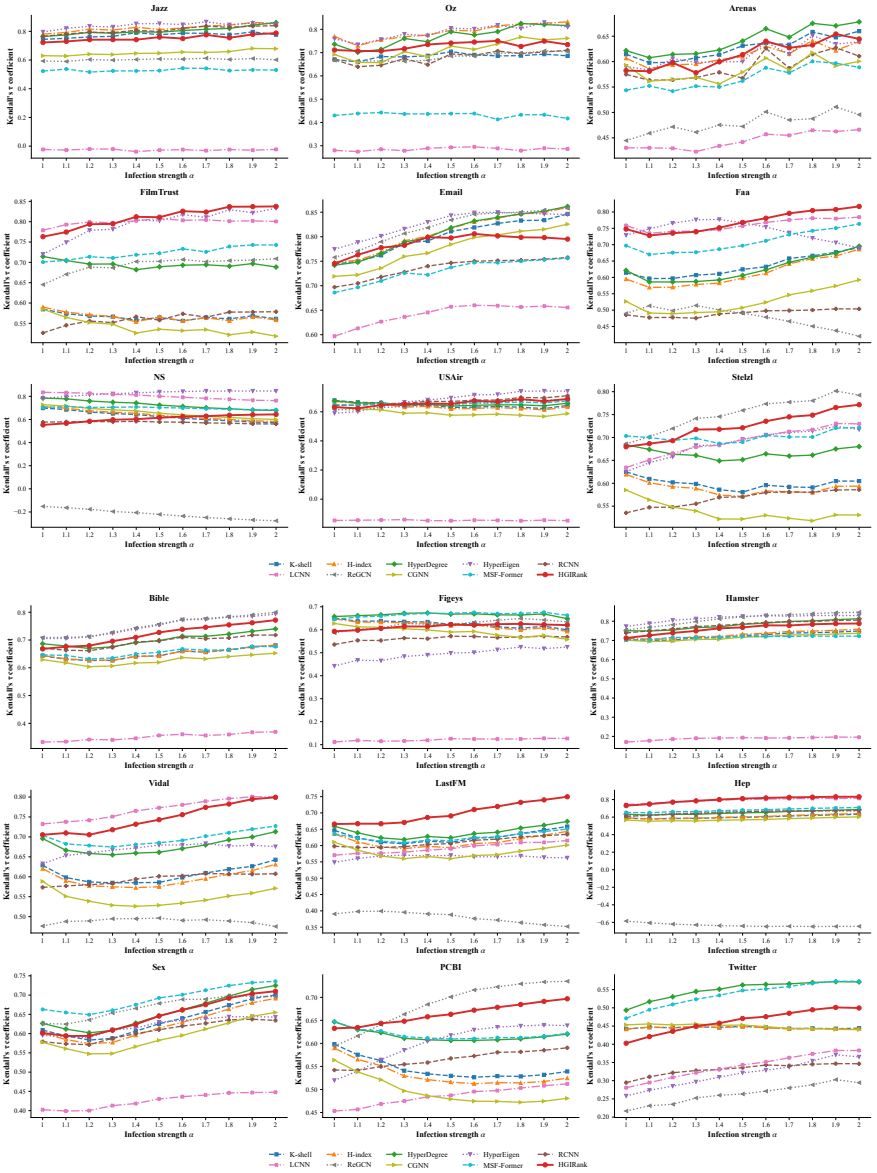


Fig. 3 Kendall's τ comparison under different infection strengths on 18 real-world networks. The upper panel reports the results on the first nine networks, and the lower panel reports the results on the remaining nine networks. Higher values indicate stronger agreement with the SIR-induced ranking.

range of cases. This observation is consistent with the overall average-rank results in Table 3, where HGIRank achieves the best comprehensive performance. A closer look at the curves also shows that different baselines are affected by network topology to varying degrees. Traditional topology-based methods may perform well when their

structural assumptions match the target network, but their curves fluctuate across networks. Learning-based baselines provide more flexibility, but their transfer performance is still not uniformly stable, especially when the target networks differ from the synthetic training graph in density, clustering pattern, and degree heterogeneity.

The stable performance of HGIRank can be attributed to its multi-view high-order structural modeling. The ego view captures local neighborhood contexts, the diffusion view describes truncated short-range expansion patterns, and the core view reflects core-shell structural roles. The adaptive reasoning module further models non-uniform node contributions, hyperedge reliability, and overlap relations among high-order structural units. These components allow HGIRank to integrate complementary structural evidence and maintain stable global ranking consistency across heterogeneous networks and varying infection-strength settings. Overall, the network-wise Kendall's τ results indicate that the performance of HGIRank is not dominated by a few specific networks. Instead, its advantage is reflected across multiple real-world networks with different structural properties, supporting the effectiveness of multi-view high-order structural reasoning for cross-network node importance ranking.

5.3 Multi-metric stability under different infection strengths

The previous subsection evaluates network-wise global ranking consistency using Kendall's τ. To further examine whether the performance of HGIRank remains stable across different evaluation criteria, we report the average results of representative methods under different infection strengths in terms of Spearman's ρ, Jaccard@10%, MAP@10%, and NDCG@10%. These metrics complement Kendall's τ by evaluating both global rank-position correlation and top-ranked node identification.

As shown in Figs. 4–7, HGIRank maintains consistently strong performance across different infection strengths and evaluation metrics. For Spearman's ρ, HGIRank achieves consistently competitive results across infection strengths, indicating stable global rank-position consistency. For Jaccard@10%, MAP@10%, and NDCG@10%, HGIRank also shows leading or competitive performance, suggesting that it can identify high-quality top-ranked nodes and place them relatively early in the predicted ranking. More importantly, the trends do not show a sharp performance degradation when the infection strength changes, which indicates that HGIRank is not overly fitted to a specific propagation setting. This stability is important because influential node rankings may vary substantially when the diffusion process becomes either too weak or too strong.

Overall, the multi-metric trends indicate that the performance of HGIRank is not limited to a single evaluation criterion. Its consistent results across global ranking correlation and top-ranked node identification suggest that multi-view high-order structural reasoning provides stable node representations under varying infection-strength settings. The results further support that the proposed model captures transferable high-order structural evidence rather than relying on a single infection-strength-specific ranking pattern.

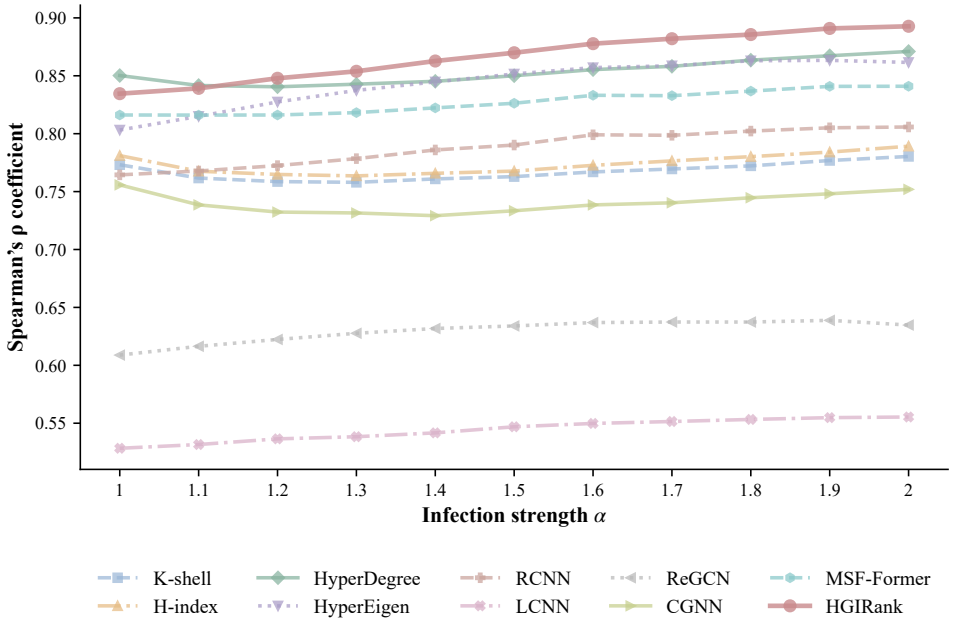


Fig. 4 Spearman's ρ under different infection strengths.

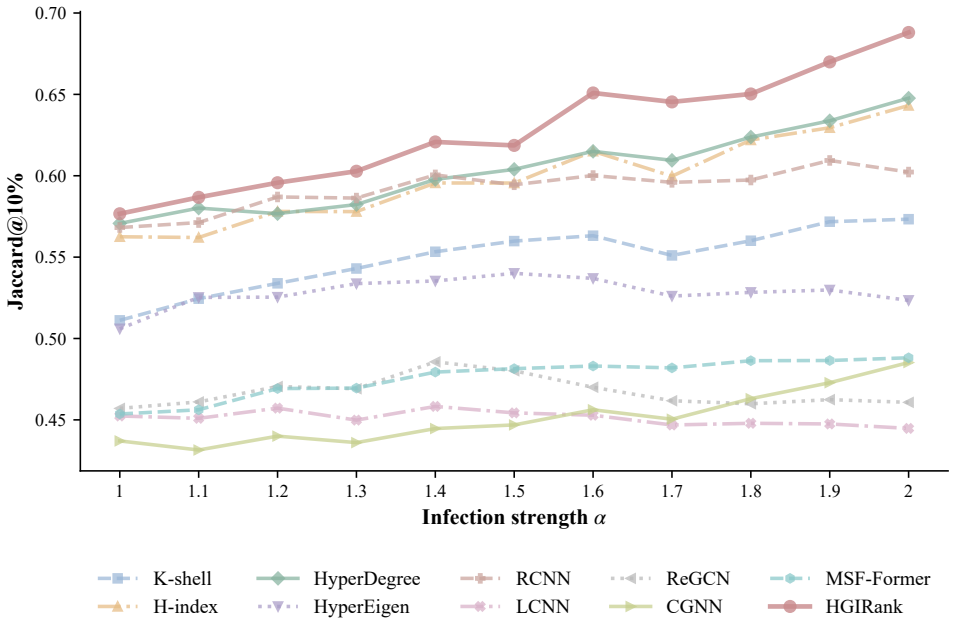


Fig. 5 Jaccard@10% under different infection strengths.

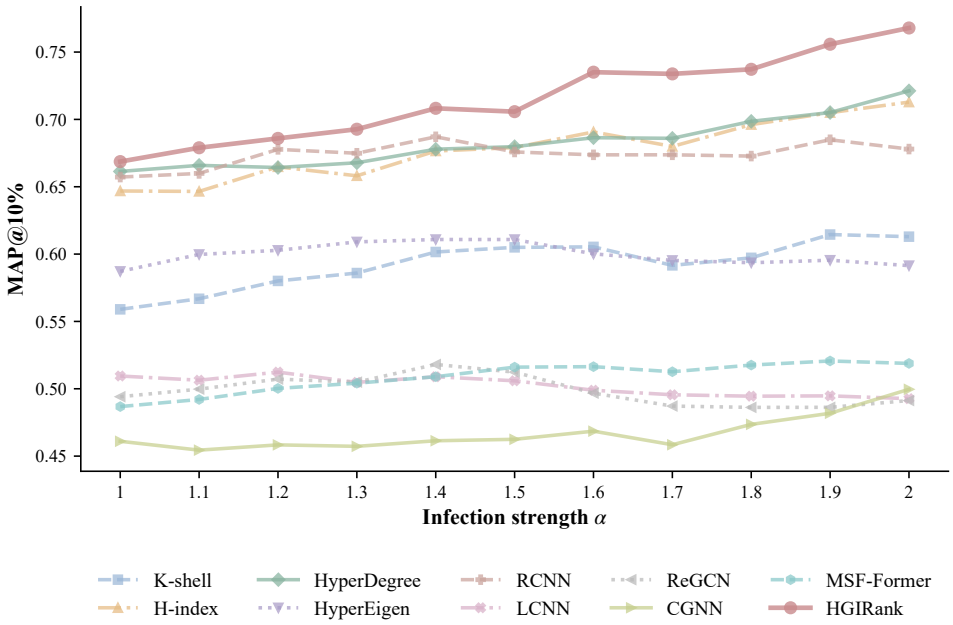


Fig. 6 MAP@10% under different infection strengths.

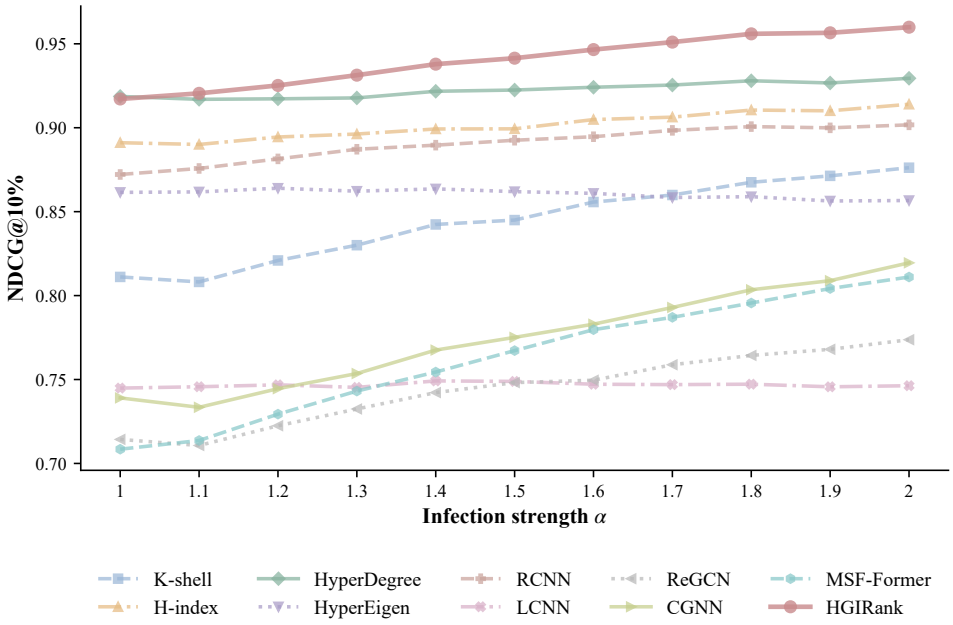


Fig. 7 NDCG@10% under different infection strengths.

5.4 Cross-network stability of Top-k node identification

In many practical scenarios, only a limited number of nodes can be selected for intervention, monitoring, or promotion. Therefore, a reliable node importance ranking method should not only achieve good average performance but also identify top-ranked nodes stably across different networks. To evaluate this property, we analyze the distribution of Top-k Rank over the 18 real-world target networks. In the boxplot, a lower median indicates better top-ranked node identification, while a narrower box reflects smaller performance variation across networks.

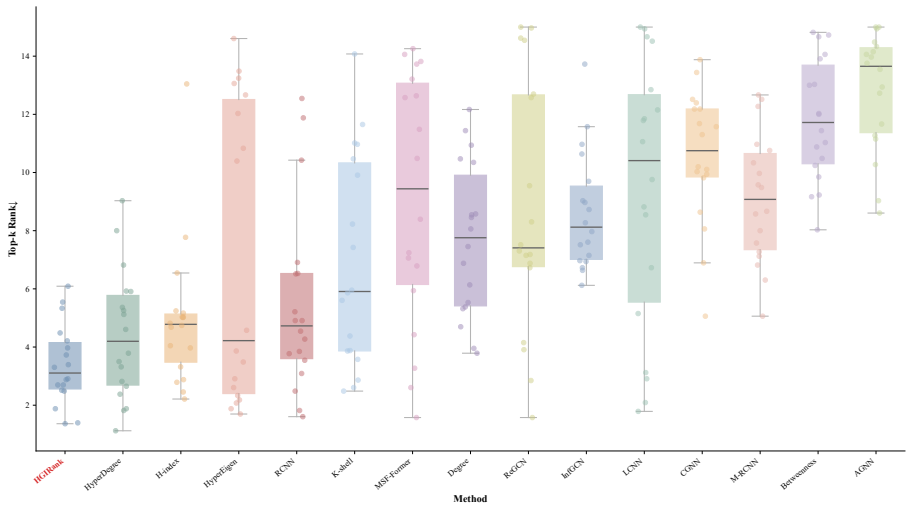


Fig. 8 Distribution of Top-k Rank across 18 real-world networks. Lower median values indicate better top-ranked node identification, while narrower boxes indicate stronger cross-network stability.

As shown in Fig. 8, HGIRank obtains a low median Top-k Rank and a relatively compact distribution, indicating that it can identify top-ranked nodes stably across heterogeneous networks. HyperDegree and H-index also achieve competitive median values, suggesting that high-order participation and local structural information are useful for top-ranked node identification. However, their distributions are generally wider than that of HGIRank, indicating that their effectiveness is more sensitive to specific network structures. Several other baselines show either higher median ranks or larger dispersion, indicating weaker cross-network stability in top-k identification.

The stable performance of HGIRank can be attributed to its multi-view high-order structural modeling. The ego view captures local neighborhood contexts, the diffusion view describes truncated short-range expansion patterns, and the core view reflects core-shell structural roles. Moreover, the adaptive reasoning module models non-uniform node contributions, hyperedge reliability, and overlap relations among high-order structural units. These components allow HGIRank to integrate complementary structural evidence, leading to both lower Top-k Rank values and stronger cross-network stability.

5.5 Ablation study

To evaluate the contribution of the main components in HGIRank, we conduct an ablation study under the same experimental setting as the main comparison. The variants are constructed by selectively enabling different hypergraph views, the adaptive node–hyperedge–hyperedge–node reasoning module, and the ranking-oriented objective used in the complete training setting. Table 4 reports the results. Here, AHR denotes adaptive high-order reasoning, and Top-k Obj. denotes the ranking-oriented top-k objective. AvgR and T/R Rank denote AvgRank and Tau/Rho Rank, respectively. Top-k Rank denotes the average rank over J@10, MAP, and NDCG. J@10, MAP, and NDCG denote Jaccard@10%, MAP@10%, and NDCG@10%, respectively. For AvgR, T/R Rank, and Top-k Rank, lower values indicate better performance, while larger values are better for the five raw metrics.

Table 4 Ablation study of HGIRank components.

Variant	Ego	Diff.	Core	AHR	Top-k Obj.	AvgR↓	T/R Rank↓	Top-k Rank↓	τ↑	ρ↑	J@10↑	MAP↑	NDCG↑
Basic-HG	✓	×	×	×	×	3.794	4.212	3.516	0.663	0.836	0.624	0.710	0.938
Diff.-only HG	×	✓	×	×	×	6.475	6.740	6.299	0.465	0.628	0.478	0.578	0.863
Core-only HG	×	×	✓	×	×	5.094	5.962	4.515	0.551	0.694	0.591	0.676	0.927
Ego+Diff. HG	✓	✓	×	×	×	4.110	3.985	4.193	0.666	0.842	0.599	0.691	0.935
Multi-view HG	✓	✓	✓	×	×	3.518	3.255	3.693	0.681	0.852	0.613	0.705	0.938
+AHR	✓	✓	✓	✓	×	2.617	2.270	**2.848**	0.693	0.861	0.626	0.715	**0.943**
HGIRank	✓	✓	✓	✓	✓	**2.392**	**1.576**	2.937	**0.699**	**0.867**	**0.628**	**0.716**	0.940

Table 4 provides several observations. First, the single-view variants show that different hypergraph views contribute differently to node importance ranking. Basic-HG, which uses only the ego view, achieves the best performance among the three single-view variants with an AvgR of 3.794. This indicates that one-hop neighborhood information provides a strong local structural signal. However, all single-view variants are inferior to Multi-view HG, suggesting that a single structural perspective is insufficient to characterize node importance across heterogeneous networks.

Second, the comparison among Basic-HG, Ego+Diff. HG, and Multi-view HG verifies the necessity of complementary view construction. Ego+Diff. HG improves T/R Rank compared with Basic-HG, but its Top-k Rank becomes worse, indicating that short-range expansion information may introduce noise when used without additional structural calibration. After the core view is included, Multi-view HG reduces AvgR from 3.794 to 3.518 and improves T/R Rank from 4.212 to 3.255. This confirms that the ego, diffusion, and core views provide complementary high-order structural contexts.

Third, adaptive high-order reasoning brings the most substantial gain among the architectural components. Compared with Multi-view HG, +AHR reduces AvgR from 3.518 to 2.617, corresponding to a relative improvement of about 25.6%. It also improves Top-k Rank from 3.693 to 2.848 and achieves the best NDCG value. These

results indicate that constructing hyperedges alone is not sufficient; it is also important to distinguish non-uniform node contributions, estimate hyperedge reliability, and propagate information among overlapping high-order structural units.

Finally, the complete HGIRank achieves the best AvgR, T/R Rank, Kendall's τ, Spearman's ρ, J@10, and MAP. Compared with +AHR, HGIRank further reduces AvgR from 2.617 to 2.392 and improves T/R Rank from 2.270 to 1.576, showing stronger overall and global ranking performance. Although +AHR obtains slightly better Top-k Rank and NDCG, the full model remains competitive in top-ranked node identification and achieves the best J@10 and MAP. This suggests that the ranking-oriented objective improves the overall balance between global ranking consistency and top-ranked node identification, rather than optimizing a single metric in isolation.

In summary, the ablation study demonstrates the complementary contributions of multi-view hypergraph construction, adaptive high-order reasoning, and the ranking-oriented training objective. The multi-view design provides complementary structural contexts, AHR extracts reliable high-order information from constructed hypergraphs, and the ranking-oriented objective further improves overall ranking quality and global ranking consistency while maintaining competitive top-ranked node identification.

6 Conclusion and future work

This paper proposed HGIRank, a multi-view hypergraph reasoning framework for influential node identification in ordinary complex networks. By constructing ego, diffusion, and core hypergraph views from an input graph, HGIRank converts local neighborhoods, short-range expansion patterns, and core-shell structural roles into complementary high-order representations. It further performs adaptive node–hyperedge–hyperedge–node reasoning to model non-uniform node contributions, hyperedge reliability, overlap-aware interactions among high-order units, and feedback from hyperedges to nodes. Extensive experiments under a unified synthetic-to-real transfer setting show that HGIRank achieves the best overall average ranking performance among 14 baseline methods across 18 real-world networks and multiple infection strengths. Ablation studies further verify the complementary contributions of multi-view hypergraph construction, adaptive high-order reasoning, and the ranking-oriented training objective. In future work, we will extend HGIRank to directed, weighted, attributed, and dynamic networks, and explore more high-order structural views for broader node importance analysis tasks.

References

Ahmad W, Wang B, Chen S (2024) Learning to rank influential nodes in complex networks via convolutional neural networks. Applied Intelligence 54:3260–3278. https://doi.org/10.1007/s10489-024-05336-x

Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512. https://doi.org/10.1126/science.286.5439.509

Battiston F, Cencetti G, Iacopini I, et al (2020) Networks beyond pairwise interactions: Structure and dynamics. Physics Reports 874:1–92. https://doi.org/10.1016/j.physrep.2020.05.004

Battiston F, Amico E, Barrat A, et al (2021) The physics of higher-order interactions in complex systems. Nature Physics 17:1093–1098. https://doi.org/10.1038/s41567-021-01371-4

Benson AR (2019) Three hypergraph eigenvector centralities. SIAM Journal on Mathematics of Data Science 1(2):293–312. https://doi.org/10.1137/18M1203031

Bick C, Gross E, Harrington HA, et al (2023) What are higher-order networks? SIAM Review 65(3):686–731. https://doi.org/10.1137/21M1414024

Bucur D (2020) Top influencers can be identified universally by combining classical centralities. Scientific Reports 10(1):20550. https://doi.org/10.1038/s41598-020-77536-7

Burges CJC, Shaked T, Renshaw E, et al (2005) Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning, pp 89–96, https://doi.org/10.1145/1102351.1102363

Chen D, Su H (2023) Identification of influential nodes in complex networks with degree and average neighbor degree. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 13(3):734–742. https://doi.org/10.1109/JETCAS.2023.3283680

Chien E, Pan C, Peng J, et al (2022) You are AllSet: A multiset function framework for hypergraph neural networks. In: International Conference on Learning Representations, URL https://openreview.net/forum?id=hpBTIv2uy_E

Duch J, Arenas A (2005) Community detection in complex networks using extremal optimization. Physical Review E 72(2):027104. https://doi.org/10.1103/PhysRevE.72.027104

Ewing RM, Chu P, Elisma F, et al (2007) Large-scale mapping of human protein–protein interactions by mass spectrometry. Molecular Systems Biology 3(1):89. https://doi.org/10.1038/msb4100134

Feng Y, You H, Zhang Z, et al (2019) Hypergraph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 3558–3565, https://doi.org/10.1609/aaai.v33i01.33013558

Freeman LC (1977) A set of measures of centrality based on betweenness. Sociometry 40(1):35–41. https://doi.org/10.2307/3033543

Freeman LC, Webster CM, Kirke DM (1998) Exploring social structure using dynamic three-dimensional color images. Social Networks 20(2):109–118. https://doi.org/10.1016/S0378-8733(97)00016-6

Gleiser PM, Danon L (2003) Community structure in jazz. Advances in Complex Systems 6(4):565–573. https://doi.org/10.1142/S0219525903001067

Guimerà R, Danon L, Díaz-Guilera A, et al (2003) Self-similar community structure in a network of human interactions. Physical Review E 68(6):065103. https://doi.org/10.1103/PhysRevE.68.065103

Guo G, Zhang J, Yorke-Smith N (2013) A novel bayesian similarity measure for recommender systems. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp 2619–2625

Ivchenko GI, Honov SA (1998) On the jaccard similarity test. Journal of Mathematical Sciences 88:789–794

Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems 20(4):422–446. https://doi.org/10.1145/582415.582418

Jiang T, Ruan Y, Yu T, et al (2025) Identifying influential nodes in complex networks via transformer with multi-scale feature fusion. Big Data and Cognitive Computing 9(5):129. https://doi.org/10.3390/bdcc9050129

Kapoor K, Sharma D, Srivastava J (2013) Weighted node degree centrality for hypergraphs. In: 2013 IEEE 2nd Network Science Workshop (NSW), pp 152–155, https://doi.org/10.1109/NSW.2013.6609212

Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: International Conference on Learning Representations

Kitsak M, Gallos LK, Havlin S, et al (2010) Identification of influential spreaders in complex networks. Nature Physics 6(11):888–893. https://doi.org/10.1038/nphys1746

Knight WR (1966) A computer method for calculating kendall's tau with ungrouped data. Journal of the American Statistical Association 61(314):436–439. https://doi.org/10.1080/01621459.1966.10480879

Kunegis J (2013) Konect: The koblenz network collection. In: Proceedings of the 22nd International Conference on World Wide Web Companion, pp 1343–1350, https://doi.org/10.1145/2487788.2488173

Kunegis J (2016) Air traffic control network dataset. KONECT – The Koblenz Network Collection, URL http://konect.uni-koblenz.de/networks/maayan-faa

Lee J, Goh KI, Lee DS, et al (2023) (k, q)-core decomposition of hypergraphs. Chaos, Solitons & Fractals 173:113645. https://doi.org/10.1016/j.chaos.2023.113645

Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data 1(1):2. https://doi.org/10.1145/1217299.1217301

Li Y, Yang X, Zhang X, et al (2022) An improved voterank algorithm to identifying a set of influential spreaders in complex networks. Frontiers in Physics 10:955727. https://doi.org/10.3389/fphy.2022.955727

Li Z, Lin Y, Wang Y, et al (2025) Critical node-aware augmentation for hypergraph contrastive learning. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, pp 5671–5679, https://doi.org/10.24963/ijcai.2025/631

Lu L, Chen D, Ren XL, et al (2016) Vital nodes identification in complex networks. Physics Reports 650:1–63. https://doi.org/10.1016/j.physrep.2016.06.007

Lü L, Zhou T, Zhang QM, et al (2016) The h-index of a network node and its relation to degree and coreness. Nature Communications 7:10168. https://doi.org/10.1038/ncomms10168

Mancastroppa M, Iacopini I, Petri G, et al (2023) Hyper-cores promote localization and efficient seeding in higher-order processes. Nature Communications 14:6223. https://doi.org/10.1038/s41467-023-41887-2

Manning CD, Raghavan P, Schütze H (2008) Introduction to Information Retrieval. Cambridge University Press, Cambridge

McAuley J, Leskovec J (2012) Learning to discover social circles in ego networks. In: Advances in Neural Information Processing Systems, pp 539–547

Miao Z, Sun F, Wang X, et al (2024) Detecting critical nodes in hypergraphs via hypergraph convolutional network. In: Web and Big Data, Lecture Notes in Computer Science, vol 14334. Springer, Singapore, pp 47–62, https://doi.org/10.1007/978-981-97-2421-5_4

Newman MEJ (2003) The structure and function of complex networks. SIAM Review 45(2):167–256. https://doi.org/10.1137/S003614450342480

Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. Physical Review E 74(3):036104. https://doi.org/10.1103/PhysRevE.74.036104

Ni X, Mei G, Zhang SS, et al (2026) Structural-aware key node identification in hypergraphs via representation learning and fine-tuning. Engineering Applications of

Artificial Intelligence 169:114108. https://doi.org/10.1016/j.engappai.2026.114108

Nieminen J (1974) On the centrality in a graph. Scandinavian Journal of Psychology 15(1):332–336. https://doi.org/10.1111/j.1467-9450.1974.tb00598.x

Ou Y, Guo Q, Xing JL, et al (2022) Identification of spreading influence nodes via multi-level structural attributes based on the graph convolutional network. Expert Systems with Applications 203:117515. https://doi.org/10.1016/j.eswa.2022.117515

Page L, Brin S, Motwani R, et al (1999) The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab

Pastor-Satorras R, Castellano C, Van Mieghem P, et al (2015) Epidemic processes in complex networks. Reviews of Modern Physics 87(3):925–979. https://doi.org/10.1103/RevModPhys.87.925

Rocha LEC, Liljeros F, Holme P (2011) Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts. PLoS Computational Biology 7(3):e1001109. https://doi.org/10.1371/journal.pcbi.1001109

Rozemberczki B, Sarkar R (2020) Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, pp 1325–1334, https://doi.org/10.1145/3340531.3411866

Rual JF, Venkatesan K, Hao T, et al (2005) Towards a proteome-scale map of the human protein–protein interaction network. Nature 437(7062):1173–1178. https://doi.org/10.1038/nature04209

Spearman C (1904) The proof and measurement of association between two things. The American Journal of Psychology 15(1):72–101. https://doi.org/10.2307/1412159

Stelzl U, Worm U, Lalowski M, et al (2005) A human protein–protein interaction network: A resource for annotating the proteome. Cell 122(6):957–968. https://doi.org/10.1016/j.cell.2005.08.029

Thang DC, Dat HT, Tam NT, et al (2022) Nature vs. nurture: Feature vs. structure for graph neural networks. Pattern Recognition Letters 159:46–53. https://doi.org/10.1016/j.patrec.2022.04.036

Wu Y, Hu Y, Yin S, et al (2024) A graph convolutional network model based on regular equivalence for identifying influential nodes in complex networks. Knowledge-Based Systems 301:112235. https://doi.org/10.1016/j.knosys.2024.112235

Xie H, Song C, Shao H, et al (2025) Hypergraph semi-supervised contrastive learning for hyperedge prediction based on enhanced attention aggregator. Entropy

27(10):1046. https://doi.org/10.3390/e27101046

Xiong Y, Hu Z, Su C, et al (2024) Vital node identification in complex networks based on autoencoder and graph neural network. Applied Soft Computing 163:111895. https://doi.org/10.1016/j.asoc.2024.111895

Yadati N, Nimishakavi M, Yadav P, et al (2019) HyperGCN: A new method for training graph convolutional networks on hypergraphs. In: Advances in Neural Information Processing Systems

Yu EY, Wang YP, Fu Y, et al (2020) Identifying critical nodes in complex networks via graph convolutional networks. Knowledge-Based Systems 198:105893. https://doi.org/10.1016/j.knosys.2020.105893

Zhang M, Wang X, Jin L, et al (2022) A new approach for evaluating node importance in complex networks via deep learning methods. Neurocomputing 497:13–27. https://doi.org/10.1016/j.neucom.2022.05.010

Zhao G, Jia P, Zhou A, et al (2020) InfGCN: Identifying influential nodes in complex networks with graph convolutional networks. Neurocomputing 414:18–26. https://doi.org/10.1016/j.neucom.2020.07.028

Zhao Y, Li C, Shi D, et al (2023) Ranking cliques in higher-order complex networks. Chaos: An Interdisciplinary Journal of Nonlinear Science 33(7):073139. https://doi.org/10.1063/5.0147721

Statements and declarations

Competing interests

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