Applied Soft Computing Journal 183 (2025) 113596

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Masked Heterogeneous Graph Attention Network for robust recommendation

Lei Sang [a], Xingwang Li [a], Yu Wang [a], Yi Zhang [a], Shun Lian [a,b], Yiwen Zhang [a,*]

[a] Anhui University, 111 Jiulong Road, Hefei, 230601, China
[b] State Key Laboratory of Cognitive Intelligence, iFLYTEK, 666 West Wangjiang Road, Hefei, 230088, China

ARTICLE INFO

Dataset link: https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding

Keywords:
Heterogeneous Graph Neural Networks
Robust recommendation
Attention-based mask
Propagation constraint probability

ABSTRACT

Heterogeneous Graph Neural Networks (HGNNs) have gained significant attraction in recommendation due to their proficiency in capturing and utilizing the diverse data types inherent in social network. Nevertheless, HGNNs are susceptible to noise and subtle adversarial attacks, as disturbances from connected nodes can cumulatively impact a target user/item node. To address this challenge, we propose the Masked Heterogeneous Graph Attention Network for Robust Recommendation (MHGAN), which aims to enhance the resilience of recommendation against adversarial attacks. Specifically, we achieve robust recommendation through two primary strategies: de-weighting and pruning. (1) **De-weighting**: We introduced meta-path based propagation constraint probability that effectively reduces the weights of perturbed edges, thereby enhancing the recommendation's robustness. (2) **Pruning**: We design an innovative attention-based masking mechanism that selectively prunes malicious neighboring nodes using topology and node features to defend against adversarial attacks. Extensive experiments on three benchmark datasets show that MHGAN achieves up to a 12% improvement in robustness under different levels of random noise compared to state-of-the-art methods HGCL and GSLRrec. Codes are available at https://github.com/lxw106/MHGAN.

1. Introduction

To address the issue of information overload on the Internet, recommendation are extensively deployed to facilitate personalized filtering [1–3]. These recommendation leverage historical behavioral records of users and items, such as clicks, purchases, ratings, and other interactions, to filter items that users may find interesting from a vast dataset [4]. Graph Neural Networks (GNNs) have recently emerged as a prevalent approach in recommendation research due to their ability to effectively model complex interactions between users and items [5–7]. The core mechanism of GNNs involves aggregating feature information from neighboring nodes on the user–item interaction graph to generate embedding of the target user/item. GNN-based recommendation can only capture local information when aggregating one-hop neighbors. However, with multi-layer propagation and aggregation, higher-order interaction information can be captured to generate more accurate user and item representations. For example, NGCF [8] enhances recommendation accuracy by propagating and aggregating embeddings over user–item interaction graphs. In comparison, LightGCN [9] improves efficiency by eliminating nonessential nonlinear activation functions and feature transformation matrices. However, in real-world recommendation scenarios, graph structures are often heterogeneous [10–12]. For example, in social recommendation [13,14], besides the interaction graph between users and items, there is also a social relationship graph among users. Effectively encoding this heterogeneous auxiliary information into the embedded representations of users and items remains a significant challenge.

In recent years, advancements in Heterogeneous Graph Neural Network (HGNN) technology have effectively addressed the auxiliary information challenge in recommendation [10,15–17]. The essence of heterogeneous graphs lies in the effective modeling of complex graph structures that involve multiple types of nodes and relationships [18], as illustrated in Fig. 1(a). HGNN-based recommendation typically employ self-attention mechanisms on subgraphs based on meta-paths [19–21] (e.g., user–movie–user) to capture specific semantic information. Additionally, to effectively integrate semantic information from different meta-paths, HGNNs assign independent parameters to each meta-path to learn and assess their importance, subsequently using these parameters to weight and compose the final embedding. For instance, HAN [22] introduces a hierarchical heterogeneous graph neural network framework that encompasses both node-level and semantic-level attention mechanisms. Node-level attention facilitates the learning of the importance of neighboring

* Corresponding author.
E-mail address: zhangyiwen@ahu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.asoc.2025.113596
Received 15 July 2024; Received in revised form 24 April 2025; Accepted 3 July 2025
Available online 23 July 2025

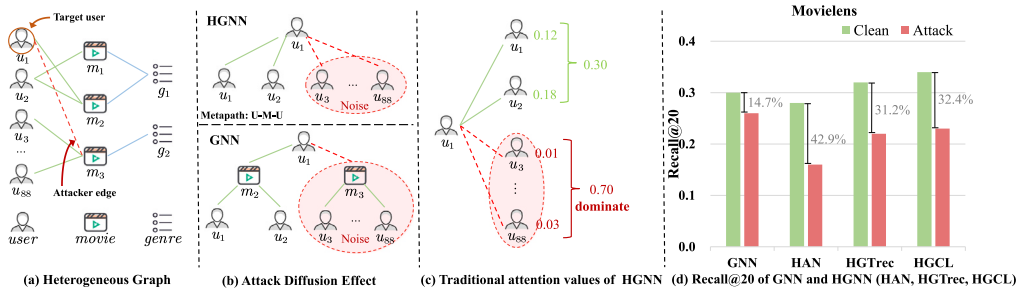

Fig. 1. An illustrative example of adversarial attacks against HGNNs on the Movielens dataset. (a) A heterogeneous graph about Movielens dataset consists three types of nodes (i.e., user, movie, genre) and two types of connections(i.e., user–movie, movie–genre). (b) The impact of GNN propagation and HGNN propagation under adversarial edge u_1-m_3. (c) An example of attention weight inertia of HGNN under adversarial edge u_1-m_3. (d) The recommended performance of GNN-based recommendation and HGNN-based recommendation under clean and attack graphs.

nodes based on meta-paths, while semantic-level attention discerns the significance of different meta-paths for recommendation.

Despite their promising performance, HGNN-based recommendation [23] are susceptible to small perturbations. Recent studies have shown that while GNNs aggregate neighbor information, they may introduce noise to the target user or item node [24–26]. Furthermore, HGNNs, due to their unique layered aggregation structure, introduce greater uncertainty in terms of robustness, necessitating further exploration and research. We assess the robustness of HGNNs by comparing the performance of HGNN-based recommendation and GNN-based recommendation under adversarial attacks [27,28] on graph structures. The results in Fig. 1(d) reveal that the performance of GNN-based recommendation decreases by 15%, whereas the performance of HGNN-based recommendation decreases by an average of 36%. This significant disparity indicates that HGNN-based recommendation are more sensitive and less robust to small perturbations. In reality, the robustness of heterogeneous recommendation systems is crucial. Taking e-commerce platforms as an example, malicious merchants may manipulate recommendation rankings by injecting false information, thereby affecting the user experience. Therefore, it is particularly important to deeply explore and effectively address the weaknesses of HGNN in terms of robustness.

In further analysis of the experimental results, we noted that the performance of HGNN-based recommendation significantly degrades when attackers connect target nodes to high-degree nodes. As an example with the HGNNs, the attacker injected an adversarial edge $u_1 - m_3$ into Fig. 1(a). The high-degree node m_3 carries a large amount of misleading information. When HGNNs uses the *'U-M-U'* meta-path to construct subgraph, which causes a large amount of misleading information to be transmitted to the target node u_1. We believe this vulnerability of the heterogeneous graph recommendation can be principally attributed to the following two factors:

(1) Attack Diffusion Effect. As shown in Fig. 1(b), in recommendation, the GNN propagates information directly in the user–movie interaction graph, and the malicious neighbor $u_3 \cdots u_{88}$ cannot have a direct impact on the target node u_1 because the interference information is weakened when passing through the one-hop neighbor m_3. However, in HGNN, the target node u_1 is directly connected to the malicious neighbors $u_3 \cdots u_{88}$ through the *'U-M-U'* meta-path, resulting in the interference information can be directly delivered to u_1. We assume that these malicious neighbors are given the same weight $1/88$ as the active nodes. At this point, the HGNN extends the impact of the attacking edges $u_1 - m_3$ to $43/44$ (i.e., the total weight of the malicious neighbors $u_3 \cdots u_{88}$).

(2) Attention Weight Inertia. The traditional attention mechanism used in HGNN-based recommendation assumes that all neighbors of a target node are beneficial and aggregates their information by assigning positive weights. However, in the example shown in Fig. 1(c), many malicious neighbors are connected to user u_1 through subgraph based on the meta-path *'U-M-U'*. Although these malicious neighbors are given small attention weights, their positive weights and large proportion allow them to dominate the receptive field [29] of the HGNN, thereby misleading the recommendation and causing inaccurate recommendations for the target user u_1.

To thoroughly address the previously mentioned vulnerabilities in HGNN, we develop a Masked Heterogeneous Graph Attention Network for robust recommendation (i.e., MHGAN). Specifically, we design a novel attention-based masking mechanism, where the mask utilizes topology and node feature to selectively prune malicious neighboring nodes to defend against adversarial attacks. More specifically, (1) to mitigate the effect of attack diffusion, we introduce meta-path based propagation constraint probability as an inherent prior estimate for MHGAN. With the prior knowledge we can **de-weight** the malicious neighbor nodes, which suppresses the spread of deceptive information and reduces the trust of malicious nodes. As illustrated in Fig. 1(b), the HGNN-based recommendations spread deceptive information to $43/44$. However, when the propagation constraint probability is introduced, the influence of deceptive information is reduced to $43X/44$ (where $X < 1$ is the constraint coefficient), which effectively suppresses the spread of deceptive information. (2) To address the defect of attention weight inertia, the mask learns a differentiable mask vector through prior knowledge and node features, and sets the attention value of the untrustworthy neighbors to 0, thus enabling the **pruning** of untrustworthy neighbors. In this way, MHGAN is able to absorb the most relevant and beneficial information. The contributions of this work are threefold:

- We have systematically investigated and assessed the robustness of HGNNs within the context of recommendation. Through our analysis, we have identified and highlighted the vulnerabilities of HGNNs to attack diffusion effect and attention weight inertia.
- We propose a novel robust recommendation which uses meta-path based propagation constraint probability as prior knowledge to de-weight malicious neighbors, limiting the spread of misleading information and reducing the impact of attacks. Additionally, it uses learned mask vectors to prune malicious neighbors, effectively mitigating the negative effects of attention weight inertia.
- We conduct extensive experiments on three real-world public datasets to evaluate the effectiveness of MHGAN. Notably, under adversarial attack scenarios, the robustness and generalization ability of our improved recommendation, MHGAN have been substantiated.

2. Preliminaries

In this section, we introduce several key concepts and definitions pertinent to HGNNs and the foundational principles of GNN-based recommendation. Table 1 provides a summary of the notation that will be utilized throughout this article.

Table 1
Notations and explanations.

Notation	Explanation
$\mathcal{G}$	Heterogeneous Graph
$\boldsymbol{\Phi}$	Meta-path
$\mathbf{P}^{\boldsymbol{\Phi}}$	Meta-path based propagation constraint probability matrix
$\mathbf{E}^{(l)}$	The matrix embeddings of the lth layer for GNN-based recommendation
$\mathbf{e}_v^0$	Node v initial embedding
$\mathbf{h}_v$	Node v projected feature
$\mathbf{W}_A$	Type-specific feature transformation matrix
$f_{vu}^{\boldsymbol{\Phi}} \in \mathbf{f}_v^{\boldsymbol{\Phi}}$	Feature-based importance of neighbor u to target node v based on the meta-path $\boldsymbol{\Phi}$
$s_{vu}^{\boldsymbol{\Phi}} \in \mathbf{s}_v^{\boldsymbol{\Phi}}$	Confidence score of neighbor u to target node v based on the meta-path $\boldsymbol{\Phi}$
$m_{vu}^{\boldsymbol{\Phi}} \in \mathbf{m}_v^{\boldsymbol{\Phi}}$	Mask value of neighbor u to target node v based on the meta-path $\boldsymbol{\Phi}$
$\hat{a}_{vu}^{\boldsymbol{\Phi}}$	Purified attentional value $\boldsymbol{\Phi}$
$\mathbf{z}_v^{\boldsymbol{\Phi}}$	Semantic embedding based on the meta-path $\boldsymbol{\Phi}$
$\gamma^{\boldsymbol{\Phi}}$	Semantic-level attention weight of meta-path $\boldsymbol{\Phi}$
$\hat{y}_{u,i}$	The interaction likelihood between user u and item i

2.1. Definition

Definition 1 (*Heterogeneous Graph.*)**.** A heterogeneous graph is a graph structure that contains nodes and edges of different types. It is represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each node is assigned a type through a function $\phi : \mathcal{V} \to \mathcal{A}$, and each edge is assigned a type through a function $\psi : \mathcal{E} \to \mathcal{R}$. $\mathcal{A}$ is the set of all possible node types, and $\mathcal{R}$ is the set of all possible edge types, with the total number of node and edge types exceeding two (i.e., $|\mathcal{A}|+|\mathcal{R}| > 2$). For each edge type $\mathbf{R} \in \mathcal{R}$, there is a corresponding binary adjacency matrix $\mathbf{M}^{\mathbf{R}}$.

Definition 2 (*Meta-path.*)**.** A meta-path $\boldsymbol{\Phi}$ is a sequence of node types and edge types in a heterogeneous graph [30], denoted as $\boldsymbol{\Phi} = \mathbf{A}_1 \xrightarrow{\mathbf{R}_1} \mathbf{A}_2 \xrightarrow{\mathbf{R}_2} \cdots \xrightarrow{\mathbf{R}l} \mathbf{A}_{l+1}$. It represents a composite relationship $\mathbf{R} = \mathbf{R}_1 \odot \mathbf{R}_2 \odot \cdots \odot \mathbf{R}_l$ between the starting node type $\mathbf{A}_1$ and the ending node type $\mathbf{A}_{l+1}$.

Definition 3 (*Meta-path based Neighbors.*)**.** In a heterogeneous graph, given a target node v and a meta-path $\boldsymbol{\Phi}$, the set of nodes that are connected to v through the meta-path $\boldsymbol{\Phi}$ is defined as the meta-path based neighbors of v, denoted as $\mathcal{N}_v^{\boldsymbol{\Phi}}$.

Definition 4 (*Meta-path based propagation constraint probability.*)**.** In this paper, we introduce the propagation constraint probabilities, denoted as $\mathbf{P}_{vu}^{\boldsymbol{\Phi}}$ into the attention mask. $\mathbf{P}_{vu}^{\boldsymbol{\Phi}}$ represents the likelihood of propagation from node v to node u along the $\boldsymbol{\Phi} = \mathbf{A}_1 \xrightarrow{\mathbf{R}_1} \mathbf{A}_2 \xrightarrow{\mathbf{R}_2} \cdots \xrightarrow{\mathbf{R}_l} \mathbf{A}_{l+1}$. By incorporating this prior knowledge, we can more effectively guide the recommendation to random walk [31] on the meta-path based subgraph, enabling it to learn high-quality node embeddings. And the meta-path based propagation constraint probability matrix $\mathbf{P}^{\boldsymbol{\Phi}}$ can be calculated by

$$\mathbf{P}^{\boldsymbol{\Phi}} = \mathbf{P}^{\mathbf{R}_1} \cdots \mathbf{P}^{\mathbf{R}_l}, \tag{1}$$

For each i within the range $1, \ldots, l$, we define $\mathbf{P}^{\mathbf{R}_i}$ as the product of the inverse of the degree matrix $\mathbf{D}^{\mathbf{R}_i}$ and the adjacency matrix $\mathbf{M}^{\mathbf{R}_i}$, expressed as $\mathbf{P}^{\mathbf{R}_i} = \left(\mathbf{D}^{\mathbf{R}_i}\right)^{-1} \mathbf{M}^{\mathbf{R}i}$. This formulation demonstrates that the probability $\mathbf{P}_{vu}^{\boldsymbol{\Phi}}$, associated with a given meta-path $\boldsymbol{\Phi}$, is characterized by two key aspects: (1) the connectivity, which reflects the quantity of paths between nodes v and u that follow meta-path $\boldsymbol{\Phi}$, and (2) the degree information, which encompasses the degrees of all nodes along meta-path.

2.2. GNN-based recommendation

Traditional GNN-based recommendation typically perform propagation and aggregation only on the user–item interaction graph [8]. We define the interaction matrix for user–item interactions as $\mathbf{S} = \mathbf{M}^{\mathbf{R}_{u-i}}$, which is the input graph for GNNs. Where $\mathbf{S} \in S^{M \times N}$, M and N denote the number of users and items, respectively. The adjacency matrix for user–item interactions can be represented as follows:

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{0} \end{pmatrix}, \tag{2}$$

In recommendation, users and items are often described using d-dimensional hidden embeddings, denoted as $e_u \in \mathbb{R}^{M \times d}$ for users and $e_i \in \mathbb{R}^{N \times d}$ for items, which can be learned from the user–item interaction graph. This can be conceptualized as transformation matrices that individually map the user/item into the hidden space, defined as $\mathbf{E} = \left[e_u; e_i\right] \in \mathbb{R}^{(M+N) \times d}$. We denote the initial embeddings for user and item as: $e_u^{(0)}$ and $e_i^{(0)}$. The way these embeddings propagation and aggregation within the user–item interaction graph is as follows:

$$\begin{aligned} e_u^{(l+1)} &= \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_i^{(l+1)}, \\ e_i^{(l+1)} &= \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e_u^{(l+1)} \end{aligned} \tag{3}$$

where $e_u^{(l)}$ and $e_i^{(l)}$ represent the embeddings of user u and item i after l-layer propagation, respectively. $\mathcal{N}_u/\mathcal{N}_i$ denotes the set of neighboring nodes for u/i. The matrix propagation form of GNNs can be defined as $\mathbf{E}^{(l)}$ that represents the matrix embeddings of the lth layer. This allows us to obtain an equivalent matrix propagation form as follows:

$$\mathbf{E}^{(l+1)} = \left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)\mathbf{E}^{(l)} \tag{4}$$

where D is a $(M+N) \times (M+N)$ Laplacian matrix. Then we take a weighted average to obtain the final user/item embeddings:

$$\mathbf{Z}_{\mathbf{user/item}} = \alpha_0\mathbf{E}^{(0)} + \alpha_1\mathbf{E}^{(1)} + \cdots + \alpha_l\mathbf{E}^{(l)} \tag{5}$$

2.3. Heterogeneous graph neural networks

HGNNs typically adopt a hierarchical aggregation strategy [22, 32]. The node-level aggregation process involves merging information from neighboring nodes on a subgraph constructed by a particular meta-path, whereas the semantic-level aggregation primarily focuses on integrating information from different meta-paths. Given target node v, its node-level embedding can be realized by the following method of aggregation:

$$\mathbf{e}_v^1 = \sigma\left(\sum_{u \in \mathcal{N}_v^{\boldsymbol{\Phi}}} \alpha_{vu}^{\boldsymbol{\Phi}} \cdot \mathbf{e}_u^0\right), \tag{6}$$

where $a_{vu}^{\boldsymbol{\Phi}}$ denotes the attention coefficient against the neighboring node u, $\mathbf{e}_v^1$ represents the node-level embedding representation of the node v obtained via the meta-path, $\mathbf{e}_u^0$ is the initial embedding state of the node u and $\mathcal{N}_v^{\boldsymbol{\Phi}}$ refers to the set of neighboring nodes of

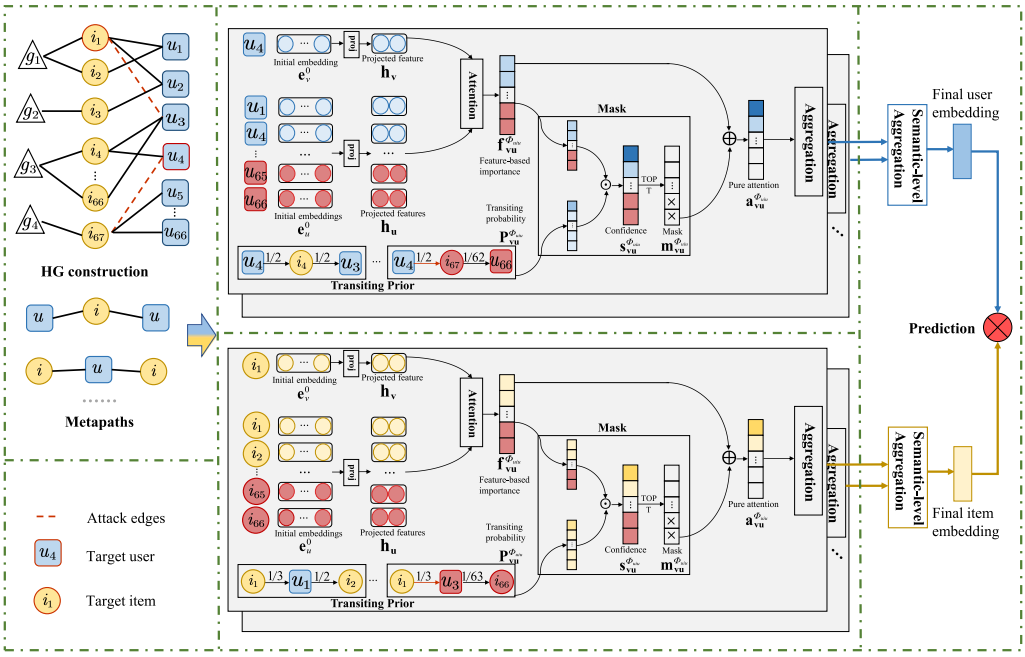

Fig. 2. The proposed MHGAN heterogeneous robust recommendation. (1) MHGAN introduces a meta-path based propagation constraint probability named transiting prior for de-weighting malicious neighbors, thus guiding the learning of embeddings for users and items. (2) MHGAN designs an innovative masked mechanism which is able to avoid the target node from aggregating malicious neighbors by pruning the edges of those untrustworthy neighbors.

node v determined based on meta-path. Given the set of meta-paths $\{\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \ldots, \boldsymbol{\Phi}_n\}$, we were able to generate n node-level embeddings, denoted as $\left\{\mathbf{e}_v^{\boldsymbol{\Phi}_1}, \mathbf{e}_v^{\boldsymbol{\Phi}_2}, \ldots, \mathbf{e}_v^{\boldsymbol{\Phi}_n}\right\}$. Then semantic level aggregation process synthesizes information from node-level embeddings obtained through each meta-path:

$$\mathbf{e}_v^2 = \sum_{j=1}^{n} \beta_{\boldsymbol{\Phi}_j} \cdot \mathbf{e}_v^{\boldsymbol{\Phi}_j}, \tag{7}$$

where $\mathbf{e}_v^2$ denotes the embedding of node v at the semantic level, and $\beta_{\boldsymbol{\Phi}_j}$ is a weight coefficient to reflect the importance of different meta-paths.

3. Proposed method

In this section, we conduct a detailed analysis of the two main shortcomings of HGNN-based recommendation and introduce our proposed MHGAN. Fig. 2 illustrates the overall architecture of MHGAN. Specifically, (1) we implement a de-weighting operation on unreliable neighbors by introducing prior knowledge, and (2) we design a mask to prune unreliable neighbors to completely eliminate the propagation of misleading information. The enhanced node-level aggregation generates high-quality user/item embeddings that are then fused into the semantic-level aggregation.

3.1. Adversarial vulnerability analysis

In this section, we delve into the phenomena observed in our adversarial attack experiments. As illustrated in Fig. 1(d), compared to GCN-based recommendation, HGNN-based recommendation, particularly the HAN, appear to be especially vulnerable in recommendation tasks. We will provide a more refined and systematic analysis of the intrinsic causes of this vulnerability.

3.1.1. Attack diffusion effect

In our study, we observe an attack diffusion phenomenon in HGNN-based recommendation that HGNN amplify the impact of adversarial high-degree nodes on other faraway nodes. As shown in Fig. 1(b), for HGNN, the impact of a high-degree node m_3 on the target node u_1 should be less than 1/2, but in practice this impact is abnormally amplified to 43/44. Specifically, when an attacker adds an adversarial high-degree node m_3 as a direct neighbor of u_1, according to the theory of network science (i.e., transfer probability), the impact of m_3 on u_1 should be proportional to the inverse of the degree of u_1 (i.e., 1/2). However, HGNN do not follow this rule and amplify the sum of influence to 43/44. This occurs because HGNNs skip intermediate layers (e.g., the item layer in a user–item interaction graph) and directly aggregate information from two-hop neighbors $u_3 \cdots u_{88}$, thereby failing to properly encode the transfer probabilities $\mathbf{P}^{ui} \cdot \mathbf{P}^{iu}$ in the structural weights of these neighbors [26].

3.1.2. Attention weight inertia

We observe that the presence of attention weight inertia may significantly weaken the generalization ability of HGNN-based recommendation in the context of adversarial attacks. As shown in Fig. 1(c), numerous malicious neighbor nodes (i.e., $u_3 \cdots u_{88}$) are able to gradually accumulate small but positive attention values. These accumulated attention values may eventually converge into a dominant effect, thereby altering the receptive domain of HGNNs and misleading the prediction scores between user and item. In light of this, the ability to assign 0 attention values to clearly untrustworthy neighboring nodes becomes particularly critical for HGNN-based recommendation.

3.2. Meta-path based propagation constraint probability

In this section, we first introduce the enhanced node-level attention mechanism of MHGAN and explain in detail how our designed mask uses meta-path based propagation constraint probability to **de-weight** malicious neighbors, addressing the attack diffusion effect in HGNN-based recommendation.

3.2.1. Node feature transformation

Given that different types of nodes carry distinct information features, these features are often distributed in different feature spaces. To process this differentiated information uniformly, the features of different node types are mapped to a common feature space. More

specifically, for a target node v of a type $A \in \mathcal{A}$, we often use the type-specific feature transformation matrix $\mathbf{W}_A$ to obtain the projected features $\mathbf{h}_v$ of the node v, as follows:

$$\mathbf{h}_v = \mathbf{W}_A \mathbf{e}_v^0, \tag{8}$$

3.2.2. Feature-based importance

Given a meta-path Φ, and under the assumption that nodes with similar features tend to be more important than nodes with dissimilar features [33,34], we use the dot product between node features to define the similarity between nodes. Accordingly, we evaluate the importance of the neighboring node u to the target node v under the meta-path Φ, denoted as f_{vu}^{Φ}:

$$f_{vu}^{\Phi} = \mathbf{h}_v \cdot \mathbf{h}_u, \tag{9}$$

In traditional node-level attention mechanisms, the feature-based importance f_{vu}^{Φ} is typically normalized directly by a softmax function on the neighbor nodes $\mathcal{N}_v^{\Phi}$ to obtain the final attention weight. However, this approach only takes into account the feature information of the node itself and ignores the graph structure. It treats the two-hop neighbors in the same way, and this equal treatment may diffuse the negative impact of those nodes with adversarial nature, as the problem pointed out in 3.1.1.

To address this issue, for the neighbour node $u \in \mathcal{N}_v^{\Phi}$, we design a differentiable masking mechanism to prune those malicious neighbor nodes with lower confidence scores s_{vu}^{Φ}. More delicately, we utilize the meta-path based propagation constraint probability $\mathbf{P}_{vu}^{\Phi}$ as an a prior estimate of the neighbor node importance, which de-weights adversarial attacking nodes and then suppresses the propagation of the negative information.

3.2.3. Propagation prior

Given a meta-path $\Phi = \mathbf{A}_1 \xrightarrow{\mathbf{R}_1} \mathbf{A}_2 \xrightarrow{\mathbf{R}_2} \cdots \xrightarrow{\mathbf{R}_l} \mathbf{A}_{l+1}$, to encode the propagation constraint probabilities along meta-path Φ as a prior estimate, we first compute the propagation constraint probability matrix $\mathbf{P}^{\mathbf{R}_i} = \left(\mathbf{D}^{\mathbf{R}_i}\right)^{-1} \mathbf{M}^{\mathbf{R}_i}$ where $R_i \in \{R_1, \ldots, R_l\}$. Each element $\mathbf{P}_{vu}^{R_i}$ in the matrix represents the propagation constraint probability from node u to node v with edge type R_i. By multiplying along the meta-path, we can compute the overall propagation constraint probability $\mathbf{P}^{\Phi}$ based on the meta-path Φ as defined in 2.1. Then, we use the element $\mathbf{P}_{vu}^{\Phi}$ in the propagation constraint probability matrix as propagation constraint of the target node v for the neighboring node u in the meta-path Φ. It is expected that if node u is indirectly connected to node v via an attacking malicious node, then the prior estimate $\mathbf{P}_{vu}^{\Phi}$ of node u as a neighbor will be small, which will address the attack diffusion effect described in 3.1.1.

3.2.4. Confidence score

To identify unreliable neighbors of the target node v, we can compute the confidence vector s_{vu}^{Φ} for neighbor nodes $\mathcal{N}_v^{\Phi}$ by combining feature-based importance f_{vu}^{Φ} and prior estimate $\mathbf{P}_{vu}^{\Phi}$.

$$s_{vu}^{\Phi} = f_{vu}^{\Phi} \cdot \mathbf{P}_{vu}^{\Phi} \tag{10}$$

where $s_{vu}^{\Phi} \in \mathbf{s}_v^{\Phi}$ and is the confidence score of neighbor $u \in \mathcal{N}_v^{\Phi}$. Neighbors with similar features and high propagation probability are considered reliable.

3.3. Differentiable mask vector

Although traditional attention mechanisms exhibits excellent differentiability properties during back-propagation [35], they do not effectively reduce the attention weights of those neighboring nodes that are malicious to 0, as demonstrated in 3.1.2. To address the problem of attention weight inertia, inspired by previous work [26], we design a differentiable masking operation. This operation effectively masks neighboring nodes with low confidence, thereby pruning malicious neighbors. Subsequently, we introduce the semantic-level aggregation of MHGAN.

3.3.1. Filter mask

For the target node v, we model the masking operation by constructing a mask vector $\mathbf{m}_v^{\Phi} \in \{0, -\infty\}^{\left|\mathcal{N}_v^{\Phi}\right|}$ for all neighbors $u \in \mathcal{N}_v^{\Phi}$. $\mathbf{m}_v^{\Phi}$ serves to control the influence of neighboring nodes on the target node v by pruning unreliable neighbors.

$$m_{vu}^{\Phi} = \begin{cases} 0 & if \quad u \in TOP\left(\mathbf{s}_v^{\Phi}, T\right), \\ -\infty & otherwise, \end{cases} \tag{11}$$

where T denotes the number of neighbor nodes we need to keep. Based on the confidence scores $\mathbf{s}_v^{\Phi}$, the $TOP(\cdot)$ operation selects T trusted neighbors and sets their mask values to 0, while the remaining neighbors are considered unreliable and set their mask values set to $-\infty$. For the unreliable neighbor node, when the softmax function is applied to the sum of the mask value $m_{vu}^{\Phi} = -\infty$ and the feature-based importance f_{vu}^{Φ} of the node v, the unreliable neighbor nodes u are pruned since the softmax function converts the $-\infty$ to 0.

Therefore, we can apply $\mathbf{m}_v^{\Phi}$ as an attentional mask to filter out unreliable neighbors and produce purified attentional values $\hat{a}_{vu}^{\Phi}$ by applying the modified softmax function:

$$\hat{a}_{vu}^{\Phi} = \frac{\exp\left(m_{vu}^{\Phi} + f_{vu}^{\Phi}\right)}{\sum_{i \in \mathcal{N}_v^{\Phi}} \exp\left(m_{vu}^{\Phi} + f_{vu}^{\Phi}\right)}. \tag{12}$$

In this way, the node-level attention mechanism is enhanced by efficiently encoding the meta-path based propagation constraint probability matrix and aggregating information only for the top T most reliable neighbors. This strategy effectively mitigates the problem of attention weight inertia.

3.3.2. Semantic-level aggregation

Finally the purified attention $\hat{a}_{vu}^{\Phi}$ is used to aggregate neighboring nodes for a specific semantic embedding $\mathbf{z}_v^{\Phi}$ denoted as

$$\mathbf{z}_v^{\Phi} = \sum_{u \in \mathcal{N}_v^{\Phi}} \left(\hat{a}_{vu}^{\Phi} \cdot \mathbf{h}_u\right). \tag{13}$$

Since different meta-paths are able to capture the varying semantic information in heterogeneous graphs, HGNNs typically employ semantic-level attention mechanism to evaluate the importance of each meta-path. For given meta-paths $\{\Phi_0, \Phi_1, \ldots, \Phi_p\}$ and target node v, after the node-level aggregation is completed, we are able to obtain a set of embeddings $\left\{\mathbf{z}_v^{\Phi_0}, \mathbf{z}_v^{\Phi_1}, \ldots, \mathbf{z}_v^{\Phi_p}\right\}$ of node v with specific semantics. Subsequently, the importance is further computed for the meta-path $\Phi \in \{\Phi_1, \Phi_2, \ldots, \Phi_p\}$ by

$$w^{\Phi} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbf{q}^T \cdot \tanh\left(\mathbf{W} \cdot \mathbf{z}_v^{\Phi} + \mathbf{b}\right), \tag{14}$$

where $\mathbf{W}$ and $\mathbf{b}$ represent the weight matrix and bias term of the MLP, respectively. The vector $\mathbf{q}$ is the attention vector used to capture the semantic-level importance. Subsequently, the importance w^{Φ} of the meta-path Φ is normalized by applying the softmax function to obtain the semantic-level attention weight γ^{Φ} of this meta-path. Finally, the final embedding $\mathbf{z}_v$ of the target node v can be obtained by semantic-level aggregation:

$$\mathbf{z}_v = \sum_{\Phi \in \{\Phi_1, \Phi_2, \ldots, \Phi_p\}} \gamma^{\Phi} \cdot \mathbf{z}_v^{\Phi}. \tag{15}$$

3.4. Computation process

To clarify the computational process, we provide a detailed explanation of the confidence score calculation and its integration with the masking mechanism, as illustrated in Algorithm 1. First, we need to calculate the propagation constraint of the target node v relative to the neighbor u, i.e., prior knowledge $\mathbf{P}_{vu}^{\Phi}$, and then we can calculate the confidence score s_{vu}^{Φ} by combining prior knowledge $\mathbf{P}_{vu}^{\Phi}$ with the feature-based importance f_{vu}^{Φ} for u. The masking mechanism is based on confidence score, and we evaluate the importance of each neighbor

node v by the confidence score, and the node u with the highest confidence score is considered more important. Then, select the top T neighbor nodes according to their confidence scores and assign their masking vector to 0. Assign the masking vector of the remaining unimportant neighbor nodes to $-\infty$. Finally, the mask vector $\mathbf{m}_{vu}^{\Phi}$ is added to feature-based importance f_{vu}^{Φ} and the purified attention value can be obtained by the softmax function. Here, because of the softmax function, $-\infty$ is converted to 0 to prune unimportant nodes, while important nodes are retained.

Algorithm 1: Confidence score based masking mechanism

Input: Meta-path Φ,
Target node v,
Neighbour node $u \in \mathcal{N}_v^{\Phi}$.
Output: Purified attention $\hat{a}_{vu}^{\Phi}$.

```
Get propagation constraint matrix,i.e., propagation prior P^Φ by Eq.(1);
for Neighbor node u ∈ N_v^Φ do
    P_vu^Φ ∈ P^Φ defines the propagation constraint between node v and u;
    Feature-based importance f_vu^Φ for u is calculated by Eq.(9);
    Get confidence score s_vu^Φ by Eq.(10);
    Filter mask vector m_vu^Φ is calculated by Eq.(11);
    Get purified attention â_vu^Φ by Eq.(12);
end
```

3.5. Optimization objectives of MHGAN

Through Semantic-level aggregation, we are able to obtain the respective final embeddings $\mathbf{z}_u$ and $\mathbf{z}_i$ for user u and item i. In our MHGAN, the interaction likelihood between user u and item i is predicted by the dot product of the user and item embeddings:

$$\hat{y}_{u,i} = \mathbf{z}_u^{\top} \mathbf{z}_i \tag{16}$$

where $\hat{y}_{u,i} \in \mathbb{R}$ denotes the prediction score of interaction likelihood between user u and item i. The larger $\hat{y}_{u,i}$ is, the higher the likelihood of interaction between the user and the item.

To better adapt our MHGAN to the recommendation task, we adopt the Bayesian Personalized Ranking (BPR) pairwise loss function [36]. Specifically, each training sample contains a user u, a positive item i^+ that the user has interacted with, and a negative item i^- that the user has not interacted with. For each training sample, we maximize the difference in prediction scores between the positive and negative samples as a means of optimizing the model, as follows:

$$\mathcal{L}_{bpr} = \sum_{(u,i^+,i^-)\in\bar{O}} -\ln\left(sigmoid\left(\hat{y}_{u,i^+} - \hat{y}_{u,i^-}\right)\right) + \lambda \|\Theta\|^2. \tag{17}$$

where λ is a hyperparameter that determines the weight of the regular term. $\bar{O} = \left\{\left(u, i^+, i^-\right) \mid \left(u, i^+\right) \in O, (u, i^-) \notin O\right\}$ contains training triples $\left(u, i^+, i^-\right)$ consisting of the interaction between the user and the item, where $\left(u, i^+\right)$ denotes the observed user–item interactions and (u, i^-) denotes the unobserved interactions. The functions $\ln(\cdot)$ and $sigmoid(\cdot)$ refer to the logarithmic and sigmoid functions, respectively. By optimizing the above BPR objective, we maximize the likelihood of the pairwise ranking relationship $\hat{y}_{u,i^+} > \hat{y}_{u,i^-}$. The overall process of our proposed MHGAN is detailed in Algorithm 2.

3.6. Time complexity analysis

For the node-level attention mechanism, MHGAN contains $|\Phi|$ meta-paths, so attention calculations need to be performed on each node's neighbors under each meta-path. The time complexity of this part is $O\left(|\Phi| \cdot E_{\Phi} \cdot d\right)$, where E_{Φ} represents the average number of edges of the subgraph built based on meta-paths, and d is the feature dimension. In the semantic-level aggregation stage, each node corresponds to $|\Phi|$ representations (one per meta-path) with a time complexity of $O(|\mathcal{A}| \cdot |\Phi| \cdot d)$, where $|\mathcal{A}|$ is the number of node types. Therefore, the overall time complexity of MHGAN is $O\left(|\Phi| \cdot E_{\Phi} \cdot d + |\mathcal{A}| \cdot |\Phi| \cdot d\right)$. Since usually $E_{\Phi} \gg |\mathcal{A}|$, the overall complexity is primarily determined by the average number of edges E_{Φ} in the subgraph. As a result, the computational efficiency of MHGAN may face challenges when handling large-scale graphs.

Algorithm 2: The overall process of MHGAN

Input: Heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
initialized features $\{\mathbf{e}_v \mid v \in \mathcal{V}\}$,
Meta-path set $\{\Phi_0, \Phi_1, \cdots, \Phi_p\}$,
Mask threshold T.
Output: Final user embeddings $\{\mathbf{z}_u \mid u \in \mathcal{V}\}$,
Final item embeddings $\{\mathbf{z}_i \mid i \in \mathcal{V}\}$.

```
Preprocessing propagation constraint matrix by Eq.(1);
for node type A ∈ 𝒜 do
    Type-specific feature transformation to obtain {h_v | v ∈ 𝒱};
end
for Φ ∈ {Φ_0, Φ_1, ⋯, Φ_p} do
    for v ∈ 𝒱 do
        Determine meta-path based neighbors N_v^Φ;
        for u ∈ N_v^Φ do
            Feature-based importance f_vu^Φ for u is calculated by Eq.(9);
            Get confidence score s_vu^Φ by Eq.(10);
            Filter mask vector m_vu^Φ is calculated by Eq.(11);
            Get purified attention â_vu^Φ by Eq.(12);
        end
        Get the node embedding z_v^Φ based on meta-path Φ by Eq.(13);
        if v equals u/i then
            z_u^Φ / z_i^Φ = z_v^Φ;
        end
    end
end
foreach Φ ∈ {Φ_0, Φ_1, ⋯, Φ_p} do Calculate the semantic-level attention γ^Φ;
Get final user and item embeddings {z_u, z_i | u, i ∈ 𝒱} by Eq.(15);
```

4. Experiments

In this section, we evaluate the robustness and effectiveness of MHGAN using three benchmark datasets. We compare the robustness of MHGAN with other HGNN-based recommendation on datasets with varying levels of noise attack. Additionally, we analyze the impact of key modules and demonstrate how MHGAN achieves robustness through enhanced node-level aggregation.

4.1. Experimental settings

4.1.1. Datasets

Three widely used datasets from different domains have been selected for the experiment: Movielens, Amazon, and Yelp. The relevant statistical information of the data has been presented in Table 2, and each dataset is described in detail as followed. **Movielens:** This dataset provides users with rating behaviors for movies, the heterogeneous relations are generated from the contained user and item side information, such as user's age attribute and movie genre information. **Amazon:** This is a large crawl of product reviews from Amazon which provides a variety of heterogeneous information about products, such as items being linked through the same brand. **Yelp:** This dataset contains heterogeneous relations (e.g., user social relations, venue rating behaviors, business attributes) in the recommendation scenario of local businesses on Yelp platform.

Table 2
Statistics of experimented datasets.

Dataset	User #	Item #	Interaction #	Sparsity
Movielens	943	1682	100 000	93.6953%
Amazon	6179	2753	195 791	98.8473%
Yelp	16 239	14 284	198 397	99.9145%

4.1.2. Baselines

In order to assess the robustness of our proposed method, we conducted a comprehensive performance comparison of MHGAN with various HGNN-based recommendation baselines. The details of the baseline models are described below:

- **Graphrec** [37]: The model is a social recommendation that contains user and item embedding learning modules as well as a rating module. User embeddings are obtained by aggregating information in item–item and user–item relationship graphs through an attention mechanism, while item embeddings are realized through item–item relationship graph aggregation. The learned embeddings are subsequently fed into the scoring module to compute the user's preference score for a particular item.
- **HAN** [22]: HAN is a graph neural network designed for heterogeneous graphs. It effectively captures and exploits the complex structural and semantic relationships among nodes by means of an attention mechanism and meta-path based information aggregation.
- **HGT** [38]: It introduces heterogeneous mutual attention for message passing scheme to refine user/item embeddings along with diverse relations in the heterogeneous graph structures.
- **HeCo** [39]: It is a self-supervised method which integrates contrastive learning with heterogeneous GNNs to consider local and high-order graph structures. Embeddings encoded with different meta-path based connections are used for contrasting.
- **SMIN** [40]: It is a self-supervised social recommendation which incorporates auxiliary graph learning task into the main task to improve the recommendation performance.
- **SMBrec** [41]: It is a self-supervised graph collaborative filtering model for multi-behavior recommendation which designs a supervised task, distinguishing the importance of different behaviors, to capture the differences between embeddings. Meanwhile, it proposes a star-style contrastive learning task to capture the embedding commonality between target and auxiliary behaviors.
- **HGCL** [10]: It proposes a Heterogeneous Graph Contrastive Learning, which is able to incorporate heterogeneous relational semantics into the user–item interaction modeling with contrastive learning-enhanced knowledge transfer across different views.
- **GSLRrec** [42]: GSLRrec can be viewed as a framework comprising two main components:graph structure learning and downstream graph recommendation tasks. The specific operational process involves optimizing the graph structure through graph structure learning to restore the graph to its pre-attack state as much as possible.

4.1.3. Hyperparameter settings

The proposed MHGAN model is implemented using the PyTorch framework. During the learning process, the embedding dimension of all entities is uniformly set to 64, and the embedding parameters are initialized using the Xavier method. We employ the Adam [43] optimizer to optimize the parameters of MHGAN. The sample batch size is determined by searching in the set $\{1024, 2048, 4096, 8192\}$. The learning rate is determined by searching in the set $\left\{1e^{-4}, 5e^{-4}, 1e^{-3}, 5e^{-3}, 1e^{-2}, 5e^{-2}\right\}$. As for the L_2 regularization coefficients λ, we search in $\left\{1e^{-6}, 1e^{-6}, \ldots, 1e^{-2}\right\}$ and found that the optimal value is $1e^{-4}$ in the usual case. In addition, we manually determined the most efficient meta-path using HAN [22] as a benchmark and consistently adopted it in other models.

4.1.4. Generating adversarial attack

In our experiments, we introduce adversarial noise edges by implementing a randomized adversarial attack. Randomized attacks are not only easy to execute but also widely applicable and commonly adopted. This was achieved by randomly selecting a certain number of edges to be removed based on the interaction history between the user and the item at specific perturbation ratios $\{10\%, 20\%, 30\%\}$. To ensure that this perturbation is not easily detectable, we also supplemented it by selecting an equal number of edges from those that have never been involved in an interaction.

4.2. Defense effectiveness of MHGAN

In this section, we evaluate the effectiveness of the MHGAN model relative to other benchmark models under two different conditions (clean and attacked). The specific experimental results are shown in Table 3. Based on these experimental results, we have the following observations:

(1) We observe that even if the attacker only attacks 10% of the edges, it is enough to significantly degrade the performance of the HAN, with the specific degradation reaching about 30%. However, MHGAN succeeds in restoring the recommended performance of HGNNs nearly to the level observed in the absence of attacks. Even with an increased percentage of attacks, the performance degradation of MHGAN is limited to about 10%, a finding that is consistent across all datasets. This robustness is primarily due to MHGAN's incorporation of a mask that efficiently differentiates between adversarial and normal edges, thereby filtering out adversarial neighbors and retaining only the critical ones.

(2) Our proposed MHGAN consistently outperforms all benchmark models in adversarial recommendation tasks.(a) HGNN-based recommendation methods (e.g., GraphRec, HAN, HGT) typically exhibit lower robustness. This is primarily because these methods fail to distinguish the importance of different neighboring nodes when facing adversarial attacks. Treating noisy nodes and positive nodes equally leads to the widespread propagation of adversarial information in the graph. MHGAN addresses this issue by introducing meta-path based propagation constraint probability to mitigate the influence of malicious neighbors, thereby suppressing the spread of deceptive information. Additionally, MHGAN employs a masking mechanism to prune untrusted neighbors by setting their attention weights to 0.(b) Recommendation based on contrastive learning [44] tend to demonstrate better robustness. This is because contrastive learning, as a form of self-supervised learning, does not rely on external labels but rather on the structure of the data itself. This learning mechanism reduces the model's dependence on the accuracy of external labeling, enabling the model to learn effectively even when the labels are of low quality. Moreover, they lack specific mechanisms to address the vulnerabilities of HGNNs, such as Attack Diffusion Effects and Attention Weight Inertia. MHGAN is specifically designed to resolve these vulnerabilities in HGNNs.

(3) Our proposed MHGAN shows excellent robustness in the face of attack. On clean graph data, MHGAN's performance is slightly inferior to GSLRrec, which may be due to MHGAN's masking mechanism, which enhances the robustness of the model by retaining only T most reliable neighbor nodes through pruning operations. However, when processing a target node with high degrees, the number of reliable neighbors may exceed the preset threshold T, resulting in some beneficial neighbors being wrongly pruned, which in turn affects the recommendation accuracy. Therefore, we need to choose the appropriate T to achieve the best balance between model performance and robustness.

4.3. Ablation study

We conducted ablation experiments to verify the necessity of the two introduced modules, propagation constraint probabilities and the filter mask, in improving the robustness and performance of the recommendation system. The details are as follows:

Table 3
Performance comparison using the datasets with 10% ∼ 30% Noisy User–item Interactions.
The best results are shown in bold and the second-best results are underlined. △ denotes the proportion of random noise.

Dataset	Model	Clean		△ = 10%		△ = 20%		△ = 30%	
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
Movielens	(2019) Graphrec	0.3037	0.3548	0.2258	0.2683	0.1702	0.2033	0.1179	0.1520
	(2019) HAN	0.2834	0.3388	0.2037	0.2364	0.1680	0.1820	0.1580	0.1658
	(2020) HGT	0.3206	0.3780	0.2865	0.3446	0.2503	0.3127	0.2172	0.2784
	(2021) Heco	0.3141	0.3786	0.2271	0.2712	0.1711	0.2102	0.1340	0.1727
	(2021) SMIN	0.2956	0.3372	0.1755	0.2031	0.1050	0.1274	0.0939	0.1084
	(2022) SMBrec	0.3240	0.3838	0.2817	0.3470	0.2732	0.3318	0.2475	0.3073
	(2023) HGCL	0.3356	0.4007	0.2898	0.3491	0.2608	0.3177	0.2320	0.2841
	(2024) GSLRrec	**0.3474**	**0.4114**	0.3061	0.3752	0.2884	0.3543	0.2759	0.3310
	MHGAN	0.3432	0.4052	**0.3397**	**0.4032**	**0.3373**	**0.4016**	**0.3348**	**0.4002**
Amazon	(2019) Graphrec	0.1180	0.0821	0.1038	0.0740	0.0897	0.0640	0.0726	0.0512
	(2019) HAN	0.1180	0.0788	0.0992	0.0677	0.0872	0.0584	0.0726	0.0479
	(2020) HGT	0.1099	0.0766	0.1028	0.0731	0.0937	0.0666	0.0869	0.0616
	(2021) Heco	0.1025	0.0704	0.0804	0.0543	0.0603	0.0415	0.0463	0.0320
	(2021) SMIN	0.1197	0.0835	0.0886	0.0621	0.0656	0.0465	0.0453	0.0312
	(2022) SMBrec	0.1246	0.0835	0.1058	0.0714	0.0928	0.0652	0.0776	0.0547
	(2023) HGCL	0.1438	0.1012	0.1285	0.0909	0.1121	0.0798	0.1068	0.0755
	(2024) GSLRrec	**0.1467**	**0.1098**	0.1302	0.0942	0.1175	0.0858	0.1083	0.0784
	MHGAN	0.1347	0.0945	**0.1321**	**0.0926**	**0.1276**	**0.0894**	**0.1205**	**0.0839**
Yelp	(2019) Graphrec	0.0752	0.0438	0.0642	0.0387	0.0628	0.0379	0.0598	0.0362
	(2019) HAN	0.0713	0.0422	0.0570	0.0335	0.0440	0.0249	0.0297	0.0168
	(2020) HGT	0.0802	0.0473	0.0613	0.0373	0.0529	0.0326	0.0496	0.0293
	(2021) Heco	0.0776	0.0483	0.0693	0.0423	0.0635	0.0388	0.0616	0.0381
	(2021) SMIN	0.0810	0.0486	0.0732	0.0443	0.0667	0.0407	0.0626	0.0389
	(2022) SMBrec	0.0848	0.0518	0.0711	0.0433	0.0660	0.0403	0.0617	0.0382
	(2023) HGCL	0.0889	0.0516	0.0762	0.0450	0.0719	0.0431	0.0677	0.0395
	(2024) GSLRrec	**0.0904**	**0.0540**	0.0801	**0.0472**	0.0748	0.0433	0.0703	0.0418
	MHGAN	0.0858	0.0505	**0.0813**	0.0420	**0.0769**	**0.0438**	**0.0744**	**0.0437**

Table 4
Ablation study on key components of MHGAN.

Data	Movielens		Amazon		Yelp	
Metric	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
w/o-P	0.3058	0.3572	0.1090	0.0747	0.0756	0.0414
w/o-F	0.3224	0.3846	0.1262	0.0869	0.0779	0.0461
w/o-PF	0.2834	0.3388	0.1180	0.0788	0.0713	0.0422
MHGAN	0.3432	0.4052	0.1347	0.0945	0.0858	0.0505

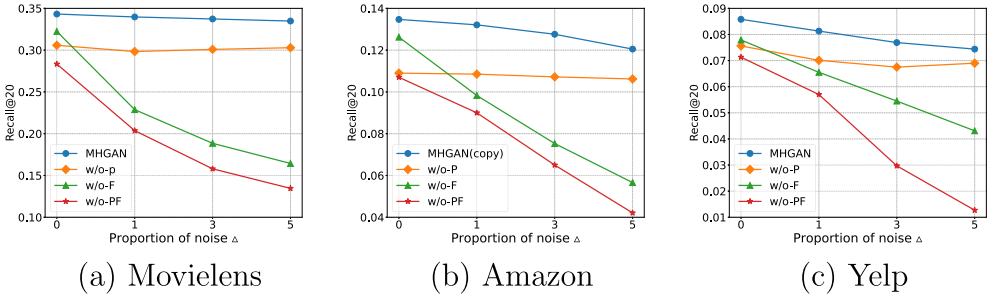

Fig. 3. Performance comparison of variants with respect to different proportions of noise on three datasets.

(1) **w/o-P:** This variant disables the **propagation constraint probabilities** in the model to facilitate the verification that our proposed **filter mask** can effectively eliminate malicious attacks on the edge.

(2) **w/o-F:** In this variant, we do not use the **filter mask** module, meaning we do not perform masking but rely solely on **propagation constraint probabilities** to enhance the model's performance and robustness.

(3) **w/o-PF(i.e.HAN):** Both **propagation constraint probabilities** and **filter mask** are simultaneously removed to assess the impact of these two modules on the model's performance and robustness.

The recommendation performance of the MHGAN framework and its variants under a clean graph is shown in Table 4. Among them, the MHGAN framework outperforms all variants, demonstrating that the constrained propagation probability and mask we introduced can effectively filter out natural noise. As shown in Fig. 3, the variant **w/o-P** exhibits the best robustness under different proportions of noise, thanks to its filter that masks out most neighbors based on feature similarity and retains only T neighbors. However, its performance is compromised by the inability to accurately mask out adversarial neighbors. Conversely, the variant **w/o-F** significantly outperforms **w/o-P** in terms of recommendation accuracy due to the introduction of prior knowledge, which guides the model to learn higher quality embeddings. Nevertheless, the equal treatment of positive and negative neighbors results in poor robustness. The variant **w/o-PF** shows the worst performance and robustness due to the absence of both modules. Based on the above analytical results, the simultaneous introduction of

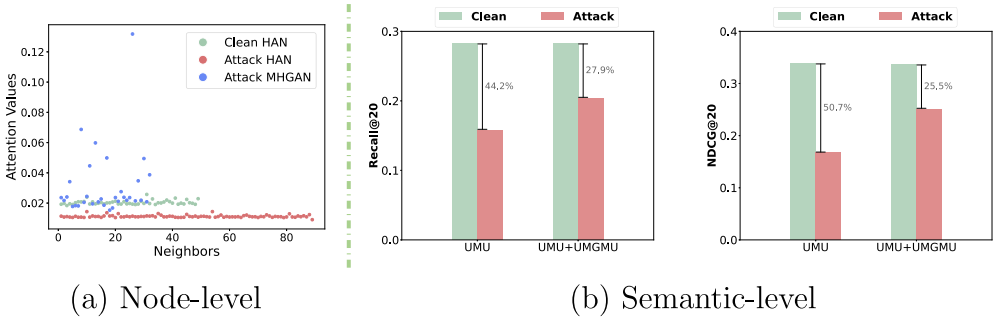

Fig. 4. Analyzing node-level and semantic-level robust aggregation. (a) Attention values about u_{855} in HAN/MHGAN model on Movielens dataset under clean/attack. (b) Results under clean/attack with different meta-paths.

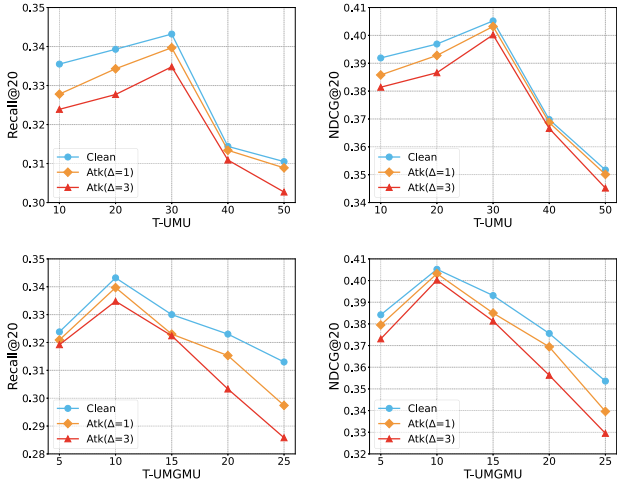

Fig. 5. Performance of different meta-paths under Clean and Attack($\triangle = \{10\%, 30\%\}$) with respect to the hyperparameter T that denotes the number of neighbor nodes we need to keep.

constrained propagation probability and the filter mask is essential to improve both the recommendation performance and robustness of the model.

4.4. Robustness of aggregations

In this section, we explore how the MHGAN model effectively shields noise neighbors by analyzing node-level attention values. Additionally, we conduct experiments using different numbers of meta-paths to examine why semantic-level attention enhances the model's robustness.

(1)Analysis of node-level aggregation. To verify that our model can effectively learn node-level attention values, we selected u_{855} in the Movielens dataset as an example. In the clean user–item interaction graph, user u_{855} has 49 neighbors in the subgraph constructed via meta-path *'U-M-U'*. Subsequently, we introduced 30% perturbations in the interaction graph, and these perturbations resulted in an additional 39 neighbors for u_{855} (i.e., 39 sources of interference for u_{855}). The attention values of the nodes are shown in Fig. 4(a). In the case of adversarial attacks, the traditional HAN model assigns positive attention values to these perturbed neighbors due to attention weight inertia, which leads to the propagation of misleading information to u_{855}, thus affecting the accuracy of the recommendation results. In contrast, the MHGAN model proposed by us effectively inhibits the spread of misleading information and significantly reduces the confidence score of perturbed neighbor nodes by introducing meta-path based propagation constraint probability. Then, the attention weights of neighbors with low confidence scores are forced to 0 through the mask machine, which not only completely blocks the negative impact of the disturbed neighbor, but also gives the trusted neighbor a higher aggregation weight through attention redistribution, so as to achieve robust and accurate information aggregation. Furthermore, there is nature noise in the original image, such as accidental clicks that can occur in an e-commerce platform, which is known as a "false positive interaction". These noises can also be effectively mitigated by the MHGAN model. This advantage is mainly due to the enhanced node-level aggregation mechanism adopted by the model. The mechanism can adaptively filter out the T neighbor nodes that are most reliable to the target node, and aggregate information from only these nodes.

(2)Analysis of semantic-level aggregation. To avoid the influence of node-level aggregation enhanced by MHGAN on the analysis of semantic-level, we selected the variant w/o-PF to focus on exploring how semantic-level aggregation enhances the robustness of heterogeneous graph neural networks. As shown in 4(b), we analyzed the performance of different meta-paths in the Movielens dataset under both clean and attack conditions. In the case of adversarial attacks, attack edges were introduced into the user–item interaction graph, i.e., attack edges were added to the meta-path *'U-M-U'*. When the variant w/o-PF used only the meta-path *'U-M-U'*, its recommendation performance significantly deteriorated due to the noise. However, when the complete meta-path was utilized, the meta-path *'U-M-G-M-U'*, which contains positive information, effectively alleviated the noise in the meta-path *'U-M-U'*, thereby significantly improving the robustness of the heterogeneous recommendation system. Moreover, even when the original graph was not attacked, the meta-path *'U-M-G-M-U'* provided additional rich semantic information, enabling the model to learn high-quality embeddings and further improving the accuracy of recommendations.

4.5. Parameter study

Masking Parameter T. We investigate the impact of the hyperparameter T (i.e., the number of neighbors retained by the mask in MHGAN) on performance in both clean and attacked environments. Taking the Movielens dataset as an example, the meta-paths used are *'U-M-U'* and *'U-M-G-M-U'*, and the interference ratio: $\{10\%, 30\%\}$. As shown in Fig. 5, as the mask threshold T increases, the model performance shows a peak, indicating that there exists an optimal value of T that yields the best performance. When the value of T is small, the mask of MHGAN can only retain less neighbor information, resulting in the inability to aggregate sufficient neighbor information, which leads to poor performance. And when the value of T is large, although the mask is able to acquire more neighbor information, the expanded receptive range also allows harmful neighbors to be included, which degrades the quality of embeddings, which negatively affects the recommendation performance. In summary, each meta-path has different characteristics for different datasets, and it is essential to choose the appropriate acceptance range for each meta-path (i.e., each

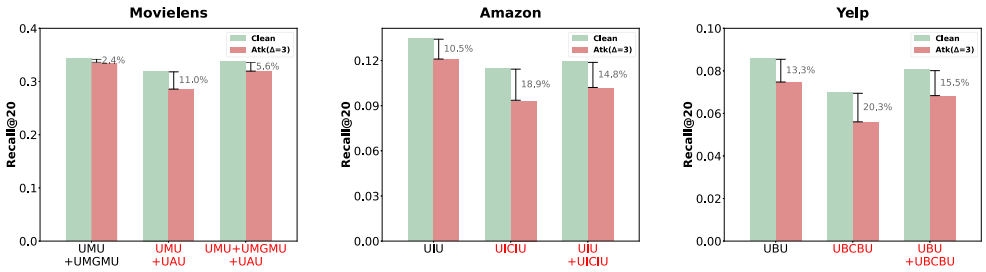

Fig. 6. Performance of MHGAN under different meta-paths. The red meta-paths are not optimal, and the black meta-path is optimal.

meta-path of each dataset needs to choose the appropriate T to allow the performance to reach the best performance).

Different meta-paths. The specific experimental results are shown in Fig. 6. Experimental results show that unreasonable meta-paths will reduce the accuracy and robustness of the recommendation system. Taking the Movielens dataset as an example, we have selected the following meta-paths for user:

- **User–Movie–User (U–M–U)**: This relationship represents user connections through commonly rated movies, which helps discover potential user groups with similar interests by analyzing co-watching patterns.
- **User–Movie–Genre–Movie–User (U–M–G–M–U)**: By incorporating movie genre information, this meta-path captures deeper semantic relationships between users' interests based on film categories, thereby improving both recommendation accuracy and system interpretability.

User's age attribute is not a direct reflection of their preference for movies. As a result, an unreasonable meta-path like "User-Age-User" can lead to incorrect user associations, which is equivalent to introducing natural noise into the data, which reduces the performance of the recommendation model. In the presence of external malicious attacks, this natural noise makes the model more susceptible to interference that weakens the robustness of the model.

5. Related work

5.1. HGNN-based recommendation

In the real world, graph structures are usually heterogeneous, so capturing rich heterogeneous information is crucial for downstream tasks. The key to heterogeneous graph neural networks lies in modeling complex graph structures involving multiple types of nodes and relationships. For example, HAN [22] achieves information propagation across different relations through a meta-path based aggregation approach that effectively captures heterogeneous information. In HGT [38], a self-attention mechanism based on the transformer module is introduced to handle complex interactions among nodes, thus enabling the model to recognize long-distance dependencies among nodes and effectively integrate information from different types of nodes and edges. In addition, MAGNN [45] optimizes the fusion between heterogeneous information by focusing on intermediate nodes on the meta-paths and carefully considering the aggregation inside and outside the meta-paths. HGIB [7] applies the information bottleneck to heterogeneous graph learning by optimizing the loss function to learn as much useful heterogeneous information as possible.

5.2. Robustness of recommendation

Recommendation rely on user–item interaction data to provide accurate recommendations to users [46]. Maliciously adding or removing interaction edges can significantly degrade the performance of recommendation. To address this challenge, Yuan et al. proposed a robust neural network framework that introduces adversarial perturbations during model training, thus enhancing the model's ability to withstand attacks [47]. Tang et al. pointed out the vulnerability of multimedia recommendation and proposed a robust recommendation model called AMR [48]. Furthermore, Deldjoo et al. introduced adversarial regularization in their model AML-RecSys to improve the stability of recommendation. Despite the success of these defense models on homogeneous graphs, they have not considered the robustness of heterogeneous graph neural networks [49]. In this paper, we propose the Masked Heterogeneous Graph Attention Network for robust recommendation to against adversarial attacks.

6. Conclusion

In this study, we examine the robustness of HGNNs in the context of recommendation. Our experimental findings reveal that HGNN-based recommendation are more susceptible to adversarial attacks compared to those based on GNNs. This vulnerability is primarily due to the attack diffusion effect and the attention weight inertia. To mitigate these issues, we introduce a robust heterogeneous graph recommendation framework, named MHGAN and improves the resilience of heterogeneous recommendations through de-weighting and pruning. Extensive experiments conducted on multiple datasets demonstrate that MHGAN significantly outperforms various other HGNN-based recommendation. As future work, MHGAN can be integrated with dynamic neural networks to handle time-evolving graph structures, thereby further improving its adaptability and robustness in real-world applications.

CRediT authorship contribution statement

Lei Sang: Writing – review & editing, Conceptualization. **Xingwang Li:** Writing – original draft. **Yu Wang:** Formal analysis. **Yi Zhang:** Supervision. **Shun Lian:** Software. **Yiwen Zhang:** Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62272001 and No. 62206002), Anhui Provincial Natural Science Foundation (2208085QF195), iFlytek Zhiyuan Higher Education Digital Transformation Innovation Research Project (2023ZY001).

Data availability

The datasets used in this paper can be accessed via this link: https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding.

References

[1] Y. Wang, Y. Zhao, Y. Zhang, T. Derr, Collaboration-aware graph convolutional network for recommender systems, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 91–101.

[2] S. Wu, F. Sun, W. Zhang, X. Xie, B. Cui, Graph neural networks in recommender systems: a survey, ACM Comput. Surv. 55 (5) (2022) 1–37.

[3] H. Chen, C.-C.M. Yeh, F. Wang, H. Yang, Graph neural transport networks with non-local attentions for recommender systems, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 1955–1964.

[4] L. Boratto, G. Fenu, M. Marras, Connecting user and item perspectives in popularity debiasing for collaborative recommendation, Inf. Process. Manage. 58 (1) (2021) 102387.

[5] C. Gao, X. Wang, X. He, Y. Li, Graph neural networks for recommender system, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 1623–1625.

[6] Y. Wang, Z. Liu, Z. Fan, L. Sun, P.S. Yu, Dskreg: Differentiable sampling on knowledge graph for recommendation with relational gnn, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3513–3517.

[7] L. Yang, F. Wu, Z. Zheng, B. Niu, J. Gu, C. Wang, X. Cao, Y. Guo, Heterogeneous graph information bottleneck, in: IJCAI, 2021, pp. 1638–1645.

[8] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 165–174.

[9] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 639–648.

[10] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, R. Luo, Heterogeneous graph contrastive learning for recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 544–552.

[11] L. Sang, Y. Wang, Y. Zhang, X. Wu, Denoising heterogeneous graph pre-training framework for recommendation, ACM Trans. Inf. Syst. (2024).

[12] L. Sang, M. Xu, S. Qian, M. Martin, P. Li, X. Wu, Context-dependent propagating-based video recommendation in multimodal heterogeneous information networks, IEEE Trans. Multimed. 23 (2020) 2019–2032.

[13] Y. Quan, J. Ding, C. Gao, L. Yi, D. Jin, Y. Li, Robust preference-guided denoising for graph based social recommendation, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 1097–1108.

[14] C. Huang, H. Xu, Y. Xu, P. Dai, L. Xia, M. Lu, L. Bo, H. Xing, X. Lai, Y. Ye, Knowledge-aware coupled graph neural network for social recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 4115–4122.

[15] S. Forouzandeh, M. Rostami, K. Berahmand, R. Sheikhpour, Health-aware food recommendation system with dual attention in heterogeneous graphs, Comput. Biol. Med. 169 (2024) 107882.

[16] S. Forouzandeh, K. Berahmand, M. Rostami, A. Aminzadeh, M. Oussalah, UIFRS-HAN: User interests-aware food recommender system based on the heterogeneous attention network, Eng. Appl. Artif. Intell. 135 (2024) 108766.

[17] M. Rostami, K. Berahmand, S. Forouzandeh, S. Ahmadian, V. Farrahi, M. Oussalah, A novel healthy food recommendation to user groups based on a deep social community detection approach, Neurocomputing 576 (2024) 127326.

[18] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, S.Y. Philip, A survey on heterogeneous graph embedding: methods, techniques, applications and sources, IEEE Trans. Big Data 9 (2) (2022) 415–436.

[19] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, Y. Li, Metapath-guided heterogeneous graph neural network for intent recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2478–2486.

[20] Y. Sun, J. Han, Meta-path-based search and mining in heterogeneous information networks, Tsinghua Sci. Technol. 18 (4) (2013) 329–338.

[21] L. Sang, W. Fei, Y. Zhang, Y. Huang, Y. Zhang, Heterogeneous adaptive preference learning for recommendation, ACM Trans. Recomm. Syst. (2024).

[22] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: The World Wide Web Conference, 2019, pp. 2022–2032.

[23] L. Sang, M. Xu, S. Qian, X. Wu, Adversarial heterogeneous graph neural network for robust recommendation, IEEE Trans. Comput. Soc. Syst. 10 (5) (2023) 2660–2671.

[24] C. Wu, D. Lian, Y. Ge, Z. Zhu, E. Chen, S. Yuan, Fight fire with fire: towards robust recommender systems via adversarial poisoning training, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1074–1083.

[25] E. Katsadouros, C. Patrikakis, A survey on vulnerability prediction using GNNs, in: Proceedings of the 26th Pan-Hellenic Conference on Informatics, 2022, pp. 38–43.

[26] M. Zhang, X. Wang, M. Zhu, C. Shi, Z. Zhang, J. Zhou, Robust heterogeneous graph neural networks against adversarial attacks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 4363–4370.

[27] A.N. Bhagoji, D. Cullina, P. Mittal, Dimensionality reduction as a defense against evasion attacks on machine learning classifiers, 2, (1) 2017, arXiv preprint arXiv:1704.02654.

[28] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13, Springer, 2013, pp. 387–402.

[29] W. Luo, Y. Li, R. Urtasun, R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 29 (2016).

[30] Y. Sun, J. Han, X. Yan, P.S. Yu, T. Wu, Pathsim: Meta path-based top-k similarity search in heterogeneous information networks, Proc. VLDB Endow. 4 (11) (2011) 992–1003.

[31] Y. Dong, N.V. Chawla, A. Swami, Metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 135–144.

[32] S. Yun, M. Jeong, R. Kim, J. Kang, H.J. Kim, Graph transformer networks, Adv. Neural Inf. Process. Syst. 32 (2019).

[33] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, T.-S. Chua, NAIS: Neural attentive item similarity model for recommendation, IEEE Trans. Knowl. Data Eng. 30 (12) (2018) 2354–2366.

[34] C.-M. Chen, C.-J. Wang, M.-F. Tsai, Y.-H. Yang, Collaborative similarity embedding for recommender systems, in: The World Wide Web Conference, 2019, pp. 2637–2643.

[35] S. Chaudhari, V. Mithal, G. Polatkan, R. Ramanath, An attentive survey of attention models, ACM Trans. Intell. Syst. Technol. (TIST) 12 (5) (2021) 1–32.

[36] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, 2012, arXiv preprint arXiv:1205.2618.

[37] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: The World Wide Web Conference, 2019, pp. 417–426.

[38] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: Proceedings of the Web Conference 2020, 2020, pp. 2704–2710.

[39] X. Wang, N. Liu, H. Han, C. Shi, Self-supervised heterogeneous graph neural network with co-contrastive learning, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1726–1736.

[40] X. Long, C. Huang, Y. Xu, H. Xu, P. Dai, L. Xia, L. Bo, Social recommendation with self-supervised metagraph informax network, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 1160–1169.

[41] S. Gu, X. Wang, C. Shi, D. Xiao, Self-supervised graph neural networks for multi-behavior recommendation, in: IJCAI, 2022, pp. 2052–2058.

[42] L. Sang, H. Yuan, Y. Huang, Y. Zhang, Graph structure learning for robust recommendation, Tsinghua Sci. Technol. (2024).

[43] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[44] L. Sang, Y. Wang, Y. Zhang, Y. Zhang, X. Wu, Intent-guided heterogeneous graph contrastive learning for recommendation, IEEE Trans. Knowl. Data Eng. (2025).

[45] X. Fu, J. Zhang, Z. Meng, I. King, Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding, in: Proceedings of the Web Conference 2020, 2020, pp. 2331–2341.

[46] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, Recommender Systems: An Introduction, Cambridge University Press, 2010.

[47] F. Yuan, L. Yao, B. Benatallah, Adversarial collaborative neural network for robust recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1065–1068.

[48] J. Tang, X. Du, X. He, F. Yuan, Q. Tian, T.-S. Chua, Adversarial training towards robust multimedia recommender system, IEEE Trans. Knowl. Data Eng. 32 (5) (2019) 855–867.

[49] Y. Deldjoo, T. Di Noia, F.A. Merra, Adversarial machine learning in recommender systems (aml-recsys), in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 869–872.