Item Relationship Graph Neural Networks for E-Commerce

Weiwen Liu, Yin Zhang, Jianling Wang, Yun He, James Caverlee, *Member, IEEE*, Patrick P. K. Chan, *Senior Member, IEEE*, Daniel S. Yeung, *Life Fellow, IEEE*, and Pheng-Ann Heng, *Senior Member, IEEE*

***Abstract*— In a modern e-commerce recommender system, it is important to understand the relationships among products. Recognizing product relationships—such as complements or substitutes—accurately is an essential task for generating better recommendation results, as well as improving explainability in recommendation. Products and their associated relationships naturally form a product graph, yet existing efforts do not fully exploit the product graph's topological structure. They usually only consider the information from directly connected products. In fact, the connectivity of products a few hops away also contains rich semantics and could be utilized for improved relationship prediction. In this work, we formulate the problem as a multilabel link prediction task and propose a novel graph neural network-based framework, item relationship graph neural network (IRGNN), for discovering multiple complex relationships simultaneously. We incorporate multihop relationships of products by recursively updating node embeddings using the messages from their neighbors. An edge relational network is designed to effectively capture relational information between products. Extensive experiments are conducted on real-world product data, validating the effectiveness of IRGNN, especially on large and sparse product graphs.**

***Index Terms*— Graph neural networks (GNNs), item relationship prediction, multihop relationships.**

I. Introduction

RECOMMENDER systems are critical for powering fast-growing web and mobile segments of the economy, to connect users to the right items (videos, jobs, news articles, and so on). These systems attempt to infer useful relationships among users and items, including user–item, user–user, and item–item relationships [1]. While there is extensive prior research on analyzing user–item interactions on the user–item bipartite graphs [1]–[4] or studying social recommendation via user–user social networks [5]–[7], item relationships in item graphs are only recently attracting attention. Discriminating item relationships between products is an essential task in e-commerce platforms by generating better and more context-relevant recommendation, as well as improving the explainability of the system [8].

On e-commerce platforms, such as Amazon, Shopify, JD.com, and Taobao, items (or products) may participate in many heterogeneous relationships like being substitutable or complementary [9]. Substitutable items are interchangeable. Typical examples include items that are viewed by the same user (*also viewed*) and items that a user who viewed one eventually bought the other (*buy after viewing*). Complementary items are usually purchased together by users. Examples include items that are *bought together* in the same transaction and items that are purchased in different transactions (*also bought*).

Understanding these item relationships can improve the accuracy and explainability of recommendation by surfacing items that are relevant to a given context. For example, when a user is browsing for headphones in an online store, recommending headphones that are cheaper or of better quality than the one she is currently viewing can better satisfy the user's needs. After she has bought a headphone, it is more reasonable to recommend her with headphone cases or chargers, rather than continuing to recommend headphones.

Foundational work in item relationship prediction has shown the potential of improving recommendation by uncovering these relationships [9]–[11]. Most existing work simply relies on item content information (reviews and descriptions) of the items to analyze the possible connections between them. However, in addition to item content features, the multihop relational information among items is also a critical part in predicting item relationships, yet is ignored by previous work. In fact, we find that there is rich potential for extracting item relationship information from multihop connections (the neighborhood of a node with path length larger than 1). The complex transitions among distant items can provide useful clues not found in the direct item connections and thus can be utilized for relationship recommendation.

Fig. 1 shows an example of predicting the relationship between i_1 and i_2, with edges representing complementary

Manuscript received March 30, 2020; revised September 18, 2020 and January 19, 2021; accepted February 11, 2021. *(Corresponding author: Pheng Ann Heng.)*

Weiwen Liu and Pheng-Ann Heng are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: wwliu@cse.cuhk.edu.hk; pheng@cse.cuhk.edu.hk).

Yin Zhang, Jianling Wang, Yun He, and James Caverlee are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: zhan13679@tamu.edu; jlwang@tamu.edu; yunhe@tamu.edu; caverlee@tamu.edu).

Patrick P. K. Chan is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: patrickchan@ieee.org).

Daniel S. Yeung, retired, was with the SMC Society of IEEE, Hong Kong (e-mail: danyeung@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3060872.

Digital Object Identifier 10.1109/TNNLS.2021.3060872

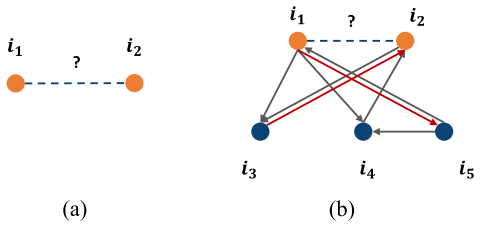


Fig. 1. Item relationship prediction. The red arrow means that the two items are substitutable. The gray arrow means that the two items are complementary. (a) Traditional item relationship prediction. (b) Item relationship prediction with multihop connections.

and substitutable relationships. Traditional methods make predictions purely according to the item content features. Such prediction can be unreliable for the cold-start items where there is no sufficient review information to infer the relationship. In contrast, rather than solely using the item content information of the two items, multihop connections exploit all the related items (i_3, i_4, and i_5) and the transitive relationship paths among them. In addition, multihop connections incorporate more item interactions so that the data sparsity problem can be alleviated to some extent.

While intuitively useful to integrate multihop connections into the item relationship prediction, it is nontrivial to preserve the complex relational dependencies in product graphs. In one direction, graph neural networks (GNNs) show promising results in aggregating information over the multihop neighborhood of nodes [12]–[14] by first constructing message of a node from its features and then propagating the message between neighbors. However, they are far from optimal for item relationship prediction. In particular, they typically focus on node features and fail to capture complex edge features. In fact, modeling edge features is one of the key tasks for item relationship prediction [8]. Moreover, current models assume that there is at most one relationship between two items, which may not be the case in real-world e-commerce applications. Two items can have multiple relationships depending on different contexts and personal perspectives. For example, two phone cases can be connected by both the *also bought* and *buy after viewing* relationships, as they may be complementary in color or texture while being substitutable in terms of function.

Therefore, in this work, we propose a novel GNN-based framework item relationship graph neural network (IRGNN), to capture the multihop relationships in the item graph. IRGNN is characterized by three unique features.

1) It incorporates multihop dependencies of relationships by recursively aggregating the information, what we wrap up into messages, from an item's neighbors.
2) We design an edge relational network to impart edge features (categorical edge types, source, and destination nodes) to the multihop message propagation process. The edge relational network learns edge-specific transformation matrices with shared network parameters and nonlinear mappings, which is more flexible and can exploit the full potentials of edge relational features.
3) We formulate the item relationship prediction problem as a multilabel link prediction task and design an outer product layer to perform the multirelationship prediction simultaneously.

The main contributions of this article are as follows.

1) We formulate item relationship prediction as a multilabel link prediction task that allows multiple relationships between items.
2) We propose a GNN-based framework, IRGNN, for explaining and predicting item relationships. IRGNN can better exploit the multihop relationships and the topological structure in the item graph.
3) We are the first to incorporate edge features for predicting the item relationship. We design a novel edge relational network to model the local, structured relational messages. The edge relational network takes as input the edge relational features and source and destination node features and outputs an edge-specific transformation matrix.
4) Extensive experiments show the effectiveness of IRGNN, especially when the graph is sparse.

II. Literature Review

This work draws on the following research areas: 1) item multirelationship-based recommendation and 2) graph-based neural networks.

A. Item Relationship Prediction

Item relationships play a significant role in user purchase decisions. Recently, discovering item relationships has received increased attention [8]–[11], [15]–[17]. Most of the existing work infers the item relationship simply from the item content information (e.g., reviews and descriptions) [8]. Sceptre [8] learns the content features of items using latent Dirichlet allocation (LDA) and fits a logistic function over the document-topic features. Chen *et al.* [18] provided personalized substitute recommendation with item-aware collaborative filtering from personalization and interpretability perspectives. Item attributes are extracted from user reviews with sentiment analysis. Linked variational autoencoder (LVAE) [15] extends Sceptre by exploiting variational autoencoders (VAEs) to avoid overfitting and producing noisy word clusters.

Another line of work seeks to use item images to infer the items' visual-level relationships. McAuley *et al.* [9] and He and McAuley [57] leveraged item images for style matching to uncover relationships at the visual level. He *et al.* [16] explored the visual information of items and proposed a mixture-of-experts framework to deal with the complex and heterogeneous item relationships. Considering the differences among item categories, Zhang *et al.* [11] aggregated both item images and descriptions in order to capture different features for heterogeneous item relationships.

Nevertheless, all the aforementioned methods consider the heterogeneous relationships with direct neighbors (single-hop neighbors), which limits the signal from multihop connections that may yield deeper insights into item relationships. SPEM [19] constructs an item copurchasing graph and predicts the substitutable relationships with a deep autoencoder to preserve first- and second-order proximities, yet fails to generate

predictions for complements and substitutes simultaneously. Wang *et al*. [10] proposed a path-constrained framework by involving two-step path constraints to infer item relationships. However, such a method may overlook useful dependencies preserved in longer paths that can help infer item relationships and is difficult to generalize to scenarios with complex and emerging relations, which limits the final recommendation performance.

Moreover, these methods assume only one single type of relationship exists between items. In practice, we find that there could be multiple relationships between items. For example, two phone cases can be connected by both the *also bought* and *buy after viewing* relationships. Thus, the assumption of only one relationship between two items would lose some of this pivotal item relationship information and further affect the final recommendation based on the predicted item relationships.

B. Graph-Based Neural Networks

Many recent research efforts have demonstrated the power of GNNs to model graph-structured data [13], [20]–[25]. For example, graph convolutional neural networks (GCNs) have achieved the state of the art in node representation for signed graphs [22], sentence classification [26], [27], and image recognition [28]. Recently, Morris *et al*. [29] built a k-dimensional GNN that can consider higher order graph structures at multiple scales. Bresson and Laurent [30] proposed a residual GNN (ConvNets), which shows that residuality can bring a significant improvement in subgraph matching and graph clustering, as they illustrated residuality could better learn multilayer architectures in complex graph-structured data.

With the growing of GNN methods, GNNs are widely applied for recommendation, such as social recommendation [31], [5], [32], [33] and knowledge graph-based recommendation [34], [35], and are even adapted to traditional recommendation methods such as collaborative filtering [36]–[38]. Recently, Fan *et al*. [5] provided a principled GNN approach with social connections and user purchase history to capture the interactions between user and items for item recommendation. Song *et al*. [31] used a dynamic graph attention network and incorporated recurrent neural networks for user behaviors in session-based social recommendation. Grad-Gyenge *et al*. [34] built a graph embedding method that took advantage of the knowledge graph to map users and items for recommendation. Considering the user–item interaction, Wang *et al*. [36] constructed a user–item interaction bipartite graph and proposed a graph-based collaborative filtering method to capture higher order connectivity in the user–item interactions. However, few of these approaches consider graph-based neural networks for item multirelationship-based recommendation.

This work is also related to link prediction (where item relationships can be viewed as missing links). Traditional link prediction methods mainly depend on heuristics to measure the proximity between nodes and infer whether the two nodes are linked, such as common neighbors and resistance distance [39], [40]. Weisfeiler–Lehman neural machine (WLNM) [41] encodes a subgraph by a fast hashing-based Weisfeiler–Lehman algorithm to automatically learn suitable heuristics for improved link prediction. Chen *et al*. [42] proposed to learn separate projections with metric learning to predict multiple relations.

Recently, GNN-based methods have shown great improvement compared with these methods. Zhang and Chen [43] developed a GNN to learn heuristics from local subgraphs for link prediction. Relational graph convolutional networks (RGCNs) [14] extended previous work by considering multiple edge types, which have a large improvement in knowledge-based data sets. Hamilton *et al*. [44] gave a review of key advancements in learning on graphs with a unified framework, including different GNN methods to embed individual nodes or (sub)graphs for link prediction or other graph tasks. HetGNN [45] is a GNN model proposed for heterogeneous graphs with recurrent neural networks and the attention mechanism. Regarding the successful performance of GNN-based models in link prediction, we build a GNN-based framework with an edge relational network to learn the complex item multirelationships.

III. Proposed Method: IRGNN

In this work, we aim at discovering heterogeneous item relationships simultaneously, such as complements or substitutes. We formulate the problem as a multilabel link prediction task. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be the directed multirelational item graph, with item feature vector $x_v \in \mathbb{R}^d$ for each item $v \in \mathcal{V}$ and edge relational vectors $e_{vw} \in \mathbb{R}^c$ for the edge $(v, w) \in \mathcal{E}$ connecting from item v to item w. Specifically, for item multirelationship-based recommendation, edge relational vectors are binary vectors, indicating the multiple relationships between item pairs. The ith element of e_{vw}, $e_{vw}^{(i)} = 1$, $i = 1, \ldots, c$ indicates that the edge from v to w has relationship i, and $e_{vw}^{(i)} = 0$ otherwise. In many real-world e-commerce applications, items are often connected by multiple types of relationships simultaneously, meaning that e_{vw} can have more than one element with value 1.

To discover item relationships, rather than having access to the full set of edges $\mathcal{E}$, we are given only a subset of edges $\bar{\mathcal{E}}$. Our task is to predict $\hat{e}_{vw}$ from v to w to recover the unknown edge relational vectors e_{vw} given any ordered pair of item feature vectors $[x_v, x_w]$.

We now present the architecture of IRGNN. The design of IRGNN is built around three research questions.

1) *RQ I:* How can we encode the local substructure features from direct neighbors into messages and aggregate the messages to update the model so that we can better preserve the topological information and the relational dependencies in the single-hop neighborhood?
2) *RQ II:* How can we better exploit the local message and propagate it to multihop neighbors to incorporate multihop relationships?
3) *RQ III:* How can we predict item relationships with the learned representation of multihop relationships in directed multirelational item graphs?

A. RQ I: Single-Hop Message Propagation

Let us denote the current node embedding of item $v \in \mathcal{V}$ by $h_v \in \mathbb{R}^d$. For the first question, we aim to design a single-hop message propagation that can encode local graph structure and relational features such as the edge type between the nodes and hidden node features into the messages, so that we can use these messages to effectively update the adjacent nodes' embeddings [46]. Only if the messages from the single-hop neighborhood are well-captured, can we further extend it to multihop neighbors. We formulate the single-hop message propagation by a message construction step, a message aggregation step, and a message update step. We propose an edge relational network to construct relational messages.

1) Message Construction: Messages from neighbors should carry information of the node embeddings of the connected neighbors and the edge relational vectors. Therefore, we define the message from item w to v as

$$m_{v \leftarrow w} = f_c(h_v, e_{wv}, h_w) \tag{1}$$

where $f_c(\cdot)$ is the message construction function and $h_v, h_w \in \mathbb{R}^d$ are the node embedding of the source and destination items. This formulation is natural since an edge in a graph is uniquely defined by the source node, destination node, and types of edges connecting them. The message construction function f_c is typically chosen to be simply a linear transformation $m_{v \leftarrow w} = W h_w$ as in [47].

Linear transformation has been shown to be effective at accumulating and encoding features from local, structured neighborhoods [14]. Based on this promising result, we design a message-specific f_c for the item relationship discovering task as

$$m_{v \leftarrow w} = g(h_w, e_{wv}, h_v) \cdot h_w \tag{2}$$

where $m_{v \leftarrow w}$ has a dimension of d, and the nonlinear function $g(\cdot)$ is our edge relational network that takes as input both the edge relational vector e_{wv} and the previous node embeddings of the source and destination nodes. It outputs a transformation matrix of dimension $d \times d$ to capture the edge relational features from h_w to h_v with relationship e_{wv}.

Edge Relational Network: In previous GNN-based models for heterogeneous graphs such as RGCN or HetGNN, a separate set of parameters is learned for each type of relationship. Applying the same transformation to the same type of relationship, regardless of which items are connected, limits the expressive power and has high computational cost when the number of types of relationships is large. Moreover, such a discrete method fails to generalize to edge features in continuous space and thus is less flexible.

To resolve the problems, we propose the edge relational network to generate an edge-specific transformation matrix. The edge relational network has the advantage of efficiently encoding complex relational information into the transformation matrices. Instead of using edge type as the single input, we propose to fully leverage the information from two connected items and take as input the edge relational vector e_{wv} and the item features h_w and h_v

$$g(h_w, e_{vw}, h_v) = \sigma\left(W_g \cdot [e_{wv} || h_w \odot h_v]\right) \tag{3}$$

where $||$ denotes the vector concatenation, $\odot$ is the elementwise product, $W_g \in \mathbb{R}^{d^2 \times (c+d)}$ is the trainable parameter, and $\sigma(\cdot)$ is the ReLU function. We reshape the output to a matrix with dimension $d \times d$.

The proposed edge relational network g preserves the edge and node features. The edge relational vector e_{wv} indicates the multirelationship between a pair of items. Motivated by the success of applying second-order feature interactions in neural networks [48], [49], rather than using a linear concatenation of h_w and h_v, we adopt a multiplicative technique $(h_w \odot h_v)$ in the edge relational network to further capture the second-order interactions of the node embeddings. Intuitively, the semantic similarity between a pair of items provides useful evidence on the item relationships. For example, two functionally similar items are likely to be substitutable. The product operation facilitates the measurement of the dependencies between the connected items. We find in experimental results in Section IV-H2 that the elementwise product can actually achieve better performance than a simple concatenation of the node embeddings.

We concatenate the edge relational vector and the product of the node embeddings to allow the edge relational network to fully exploit the local structure and relational information. Though multiple message propagation explained in Section III-B, messages captured by the edge relational network can be propagated along the paths and improve the item relationship prediction.

2) Message Aggregation: Then, we aggregate messages from the item's local neighborhood with a *mean* aggregator by averaging messages from neighbors

$$m_v = \frac{1}{|\mathcal{N}_v|} \sum_{w \in \mathcal{N}_v} m_{v \leftarrow w} \tag{4}$$

where $\mathcal{N}_v$ denotes the set of direct neighbors of item v and $|\mathcal{N}_v|$ is the number of neighbors of item v. This *mean* aggregator is nearly equivalent to the convolutional propagation rule used in the transductive GCN [47]. One could use more complex aggregators, such as LSTMs [50] or attention mechanisms [51].

3) Message Update: After obtaining the messages m_v from the neighborhood, we use the previous node embedding of itself and newly received neighborhood messages to update the model.

In general, the updated $h'_v \in \mathbb{R}^d$ is expressed as

$$h'_v = f_u(h_v, m_v) \tag{5}$$

where $f_u(\cdot)$ is a nonlinear function.

In experiments, we adopt a simple update scheme

$$h'_v = f_r(h_v, m_v) + h_v \tag{6}$$

$$= \sigma(W_1 \cdot [\sigma(W_2 h_v) || m_v]) + h_v \tag{7}$$

by first mapping the previous node embedding into the message space and then concatenate it with the current message and feed to a second fully connected layer. The weight matrices $W_1 \in \mathbb{R}^{d \times 2d}$ and $W_2 \in \mathbb{R}^{d \times d}$ are model parameters. A gated recurrent unit (GRU) [52] is introduced as the message update function for small-scale graphs, such as molecular

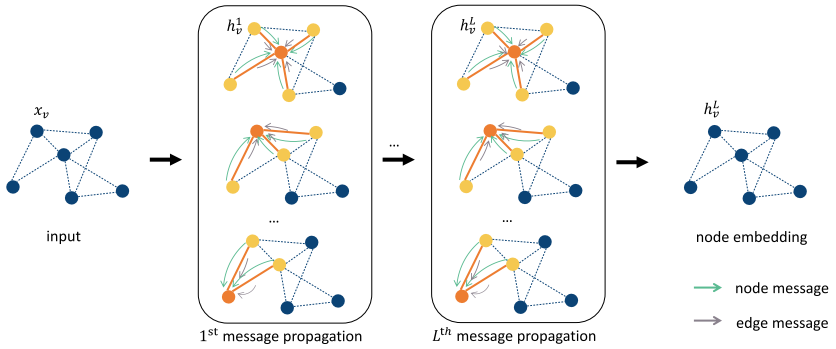


Fig. 2. Illustration of the message propagation process.

graphs with $f_u = \text{GRU}(h_v, m_v)$ [46], [53]. Though the gated mechanism can use information from the neighboring nodes and the previous time step to update the node embedding, it is computationally expensive, especially when the graph is large. In our experiments (see Section IV-H2), we show that two fully connected layers with a shortcut connection as in (7) can achieve similar or even better performance as a GRU for item relationship prediction. Details of the shortcut connection will be discussed in the following.

B. RQ II: Multihop Relationships

With the node embedding representing messages from each item's single-hop neighbors, we are ready to propagate the messages across the graph and model multihop relationships. Multihop connections preserve relational dependencies among distant items and provide crucial evidence for inferring item relationships.

We stack L message propagation steps together and recursively aggregate messages from neighboring nodes, as shown in Fig. 2, so that the messages containing relational and structured information from a node's L-hop neighbors can be explicitly encoded to the node embeddings. We initialize $h_v^{(0)} = x_v$. Therefore, for $l = 1, \ldots, L$, our message propagation step is recursively expressed as follows:

$$m_v^{(l+1)} = \frac{1}{|\mathcal{N}_v|} \sum_{w \in \mathcal{N}_v} \sigma\left(W_g^{(l)} \cdot [e_{vw} || h_w^{(l)} \odot h_v^{(l)}]\right) \cdot h_w^{(l)} \tag{8}$$

$$h_v^{(l+1)} = \sigma\left(W_1^{(l)} \cdot \left[\sigma\left(W_2^{(l)} h_v^{(l)}\right) \middle|\middle| m_v^{(l+1)}\right]\right) + h_v^{(l)} \tag{9}$$

where $h_v^{(l)} \in \mathbb{R}^d$ and $m_v^{(l)} \in \mathbb{R}^d$ denote the node embedding and the message of item v at the lth message propagation step (layer). We use different parameters $\{W_g^{(l)}, W_1^{(l)}, W_2^{(l)}\}$ in different layers to extract hierarchical latent features. As such, multiple message propagation steps seamlessly inject multihop information into the node embeddings.

a) Shortcut connections: To facilitate IRGNN to learn deeper models to involve neighbors from farther away, we use shortcut connections [54] between hidden layers. Shortcut connections enable the model to carry over information from the previous iterations, with the added benefit of not involving any extra parameters or computational complexity.

Without a shortcut connection, the first term $f_r(h_v, m_v)$ in (6) should directly learn the representation of h'_v, but with an identity mapping h_v, $f_r(h_v, m_v)$ only needs to learn residual mapping

$$h'_v - h_v. \tag{10}$$

With the increase of the number of iterations, if no new residual mapping is needed, the network can bypass identity mappings, which could greatly simplify the training for learning multihop relationships.

C. RQ III: Item Multirelationship Prediction and Optimization

After obtaining the embeddings, we propose an outer product layer to reconstruct the given graph structure and perform the item multirelationship recommendation.

1) Outer Product: Given an ordered pair of learned node embedding $[h_v^{(L)}, h_w^{(L)}]$ at the last step L, our prediction $\hat{e}_{vw}^{(i)}$ for relationship i from v to w, $i = 1, \ldots, c$ is

$$\hat{e}_{vw}^{(i)} = \mu\left(W_o^{(i)} \cdot \text{vec}(h_v^{(L)} \otimes h_w^{(L)})\right) \tag{11}$$

where $\mu(x) = (1/1 + e^{-x})$ is the sigmoid function. Every relationship i is associated with a weight vector $W_o^{(i)} \in \mathbb{R}^{d^2}$. We vectorize the outer product $h_v^{(L)} \otimes h_w^{(L)}$ to dimension d^2, allowing the model to fully exploit underlying feature interactions. Moreover, the outer product operation is noncommutative so that $[h_v^{(L)}, h_w^{(L)}]$ and $[h_w^{(L)}, h_v^{(L)}]$ yield different results. As such, we can make predictions with directions.

2) Optimization: We optimize the sum of the cross-entropy loss for each type of relationship separately, to push the model to score observable relationships higher than the negative ones

$$\mathcal{L} = \sum_{(v,w) \in \mathcal{T}} \sum_{i=1}^{c} e_{vw}^{(i)} \log\left(\hat{e}_{vw}^{(i)}\right) + \left(1 - e_{vw}^{(i)}\right) \log\left(1 - \hat{e}_{vw}^{(i)}\right) \tag{12}$$

where $\mathcal{T}$ is the training edge set. However, it is computationally consuming to train GNNs on large-scale graphs [55]. We adopt the following two methods to advance the training efficiency while preserving the accuracy.

1) *Negative Sampling:* Since it is impractical to use all nonedges as negative samples, we train the model with negative sampling. In the multigraph setting, item pairs can have multiple different relationships. Treating samples with one type of relationship as negative samples of the ones with the other type as in [8] is not applicable.

Therefore, we randomly sample a set of nonedges $\bar{\mathcal{E}}$ as many as there is at least one type of relationships ($|\mathcal{E}| = |\bar{\mathcal{E}}|$).

2) *Neighborhood Sampling:* Most existing GNN-based methods require that all nodes in the graph are present during the training of the embeddings, which is time-consuming and not applicable when the graph is large [56]. Take the Amazon product graph for example, and there are thousands of nodes and millions of edges, making it infeasible to directly operate on the full graph. To scale up the model, we adopt a uniform neighborhood sampling strategy [23], [56] by sampling the neighborhood around a node and dynamically constructing a computation graph containing all the nodes needed for L iterations from this sampled neighborhood. Therefore, our model can be trained in a minibatch mode.

D. Discussion

We conclude our presentation of the IRGNN model with a discussion of its relation to other models and time complexity analysis.

1) Relations to Other Models: IRGNN is a general framework for item relationship prediction and can be used to learn any type of item relationship. We can show that most existing models can be viewed as special cases of IRGNN.

a) Relation to LVAE: LVAEs is the state-of-the-art model for predicting relationships between products [15]. Since LVAE considers direct neighbors only, we can set the number of message propagation iterations in IRGNN to $L = 1$, yielding the deterministic version of LVAE. The message construction is

$$m^{(1)}_{v\leftarrow w} = \left[W_a h^{(0)}_v + s_a \middle|\middle| W_b h^{(0)}_w + s_b\right]$$

by feeding the source and destination node features to a fully connected neural networks separately, where $\{W_a, W_b, s_a, s_b\}$ are model parameters. We use a mapping $m^{(1)}_v(w) = m^{(1)}_{v\leftarrow w}$ as the aggregator. The message update function is a simple elementwise ReLU function $\sigma(\cdot)$

$$h^{(1)}_v(w) = \sigma\left(m^{(1)}_v(w)\right).$$

LVAE uses a fully connected layer to predict the relationships

$$\hat{e}^{(i)}_{vw} = W_c \cdot h^{(1)}_v(w) + s_c$$

where $\{W_c, s_c\}$ are parameters to be learned.

b) Relation to RGCN: RGCNs is the state-of-the-art GNN-based model developed specifically to deal with the multirelational data [14]. We can view the RGCN message construction function as

$$m^{(l+1)}_{v\leftarrow w} = \sum_{i=1}^{c} \frac{1}{c_{v,i}} e^{(i)}_{wv} W^{(l)}_i h^{(l)}_w$$

where $c_{v,r}$ is a predefined normalization constant. The transformation matrix $W^{(l)}_i$ remains the same for relationship i at layer l, which is less expressive than our designed edge relational network. Then, the message is aggregated by a *Sum* aggregator $m^{(l+1)}_v = \sum_{w\in\mathcal{N}_v} m^{(l+1)}_{v\leftarrow w}$. The node embeddings are updated by the message update function $h^{(l+1)}_v = \sigma(m^{(l+1)}_v + W^{(l)}_0 h^{(l)}_v)$.

In sum, IRGNN is a flexible framework for item relationship prediction. IRGNN generalizes the existing methods by incorporating multihop relationships with an edge-dependent edge relational network.

2) Time Complexity Analysis: One limitation of GNN-based models is scalability. As we use the neighborhood sampling to break down the graph into minibatches of subgraphs, the complexity of a single step of the message propagation for a dense graph is reduced from $O(n^2d^2)$ to $O(m^2d^2)$, where n is the number of nodes of the full graph, $m \ll n$ is the number of nodes in each batch, and d is the dimension of the node embedding. Therefore, the overall time complexity is $O(m^2d^2L)$ with L message propagation steps.

Empirically, IRGNN trains a graph with thousands of nodes and millions of edges within 300 s per epoch on a TITAN Xp Graphics Card, following the setting in Section IV-B. As for prediction, IRGNN predicts millions of edges at the same scale within 30 s.

IV. Experiments

In this section, we evaluate the proposed IRGNN on several Amazon-based data sets and a Taobao data set. We aim to answer the following research questions.

1) *RQ1:* How does IRGNN perform as compared with the state-of-the-art item relationship prediction models?
2) *RQ2:* How well does IRGNN perform on sparse data as compared with the state-of-the-art model?
3) *RQ3:* How do different hyperparameter settings (the number of message propagation iterations and the dimension of node embeddings) influence the performance of IRGNN?
4) *RQ4:* How does each designed component (the shortcut connection, the edge relational network design, the message update scheme, the outer product, the neighborhood sampling, and the negative sampling) influence the performance of IRGNN?
5) *RQ5:* How does IRGNN benefit the recommendation in e-commerce platforms with improved accuracy and explainability?

A. Data Set Description

1) Amazon Data: We use the Amazon data set from [9], [57]. The complete data set contains over 1 million products and 42 million copurchase relationships across around 20 top-level product categories. We focus on five main categories that display different complementary aspects: Video Games, Musical Instruments, Movies and TV, Electronics, and Clothing, Shoes, and Jewelry. Each category forms a separate item graph. The data set contains four different types of relationships and all of them are asymmetric.

1) Also Bought (AB): Users bought x also bought y.
2) *Also Viewed (AV):* Users viewed x also viewed y.
3) *Bought Together (BT):* Users frequently bought x and y (x and y were purchased as part of a single basket).

TABLE I
STATISTICS OF THE AMAZON DATA

categories	#items	#edges	#multi-edges	density (10^{-5})	#AB	#AV	#BT	#BV
Video Games	43,592	1,458,730	95,357	5.02	1,143,763	170,107	27,460	117,400
Musical Instruments	58,866	1,156,946	84,180	2.43	531,379	480,710	26,955	117,902
Movies & TV	138,996	3,244,921	230,124	1.19	2,766,430	172,940	80,924	224,627
Electronics	333,488	6,269,914	513,521	0.46	2,550,227	2,823,653	126,166	769,868
Clothing, Shoes, and Jewelry	615,842	8,275,321	982,646	0.26	2,188,897	5,875,987	208,744	1,693

4) *Buy After Viewing (BV):* Users who viewed x eventually bought y.

We remove the isolated nodes and use the reviews as the node feature, following the preprocessing steps in [15]. The detailed statistics of the data are shown in Table I.[1]

a) Mutigraph and simple graph setting: As discussed in Section II, existing work on item relationship recommendation assumes only one single type of relationship exists between items [8]–[11], [15]–[17]. From Table I, we can observe that items are actually connected by multiple relationships (multiedges). Following the previous setting would lose some pivotal item relationships and influence the recommendation results. Though one could formulate the problem as a simple graph setting with $2^4 = 16$ different types of edges, the size of the parameters of the model will increase exponentially, and the limited training data for each type of edge will result in poor prediction performances.

Moreover, *also bought* relationship dominates the graph in Video Games, Musical Instruments, and Movies & TV, whereas *also viewed* relationship is the dominant one in Clothing, Shoes, and Jewelry. Discriminating relationships on the imbalanced data with one dominant relationship is easier in the simple graph setting since the goal is to *differentiate* between edge types. In contrast, the setting of the multigraph is more challenging due to the more relaxed assumption.

2) Taobao Data: Taobao data is a data set consisting of user behavior data retrieved from Taobao,[2] one of the biggest e-commerce platforms in China, with 987 994 users, 4 162 042 items, and 100 150 807 interactions. It contains user behaviors from November 25 to December 3, 2017, with several behavior types, including click, purchase, adding to cart, and item favoring. We assume that an item relationship exists if more than 50 users perform the same behaviors toward a pair of items, and thereby, the Taobao product graph has the following four types of relationships.

1) *Coclick:* Users clicked x also clicked y.
2) *Copurchase:* Users bought x also bought y.
3) *Add-both-to-cart:* Users added x to the shopping cart also added y.
4) *Favor-both:* Users favored both x and y.

Node features are item ID and the corresponding category ID. Only items that have at least 50 interactions are kept. After removing the isolated nodes, the retained item graph contains 224 654 nodes and 2 410 056 multiedges.

[1]Note that the number of BV of clothing, Shoes, and Jewelry in Table I is much smaller. It may be due to the crawling strategy of the data set that products with BV are rarely sampled. The preprocessing procedure also removed some products with no review information.

[2]https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

B. Experimental Setting

Our code is available at https://github.com/wwliu555/IRGNN_TNNLS_2021. We randomly assign paired item samples into the training, validation, and test sets with an 8/1/1 ratio. The number of relationships $c = 4$. We use grid search to select the hyperparameter for all the methods on the validation set: the node embedding size $d \in \{8, 16, 64, 128\}$, learning rate in $\{0.001, 0.01, 0.1\}$, the batch size is set to 512, and the optimization method is Adam [58]. For our proposed IRGNN, the message propagation iterations $L \in \{2, 3, 4, 5\}$ and $d \in \{8, 16, 32\}$, and we select $d = 16$ for all the experiments. All models are trained 300 epochs to ensure convergence. We conduct five independent runs and use early stopping. For a fair comparison, the input node features x_v are the same, followed by a fully connected embedding layer mapping raw node features from the original dimension to the dimension of the node embeddings d.

C. Baselines

1) *Logistic Regression (LR):* Our first baseline is a straightforward application of LR. We first feed the node feature into an embedding layer to obtain the node embeddings and then concatenate the pair of node embeddings $[h_v||h_w]$ as the input of four logistic models, one for predicting one type of relationship.
2) *Sceptre:* Sceptre fits a logistic classifier over the topic space of LDA, which not only learns the relationship between items but also the direction of the relationship [8].
3) *PME:* PME [42] incorporates multirelations by embedding separate different node and edge types into different latent spaces and uses metric learning to capture the first- and second-order proximities.
4) *PMSC:* PMSC is a path-constrained method to discriminate substitutes and complements (PMSC) [10]. Specifically, PMSC incorporates 2-hop path constraints with t-norm fuzzy logics.
5) *RGCN:* RGCNs is a GNN-based model developed specifically to deal with the multirelational data [14]. RGCN handles different types of relationships and directions separately and uses a weighted sum to aggregate them.

TABLE II
EXPERIMENTAL RESULTS ON AMAZON DATA

		LR	Sceptre	PME	PMSC	RGCN	HetGNN	LVAE	**IRGNN**	imp%
Video Games	ACC	0.7293	0.7550	0.7651	0.7515	0.7608	0.7873	0.8175	**0.8403***	+2.79%
	Precision	0.6923	0.7395	0.6743	0.7458	0.6889	0.7565	0.8169	**0.8177**	+0.10%
	Recall	0.6391	0.6653	0.6950	0.6318	0.6421	0.7340	0.7479	**0.8137***	+8.80%
	AUC	0.8615	0.8899	0.8762	0.8848	0.896	0.9112	0.9367	**0.9631***	+2.82%
Musical Instruments	ACC	0.7083	0.7329	0.7468	0.7191	0.7481	0.7806	0.7925	**0.8129***	+2.57%
	Precision	0.7356	0.7120	0.7605	0.7369	0.7578	0.7723	0.7976	**0.7977***	+0.01%
	Recall	0.5485	0.6784	0.6465	0.5679	0.6120	0.6948	0.7109	**0.7546***	+6.15%
	AUC	0.8785	0.8869	0.8972	0.8963	0.9146	0.9293	0.9417	**0.9505***	+0.93%
Movies and TV	ACC	0.692	0.7398	0.7511	0.7438	0.7677	0.7657	0.8065	**0.8322***	+3.19%
	Precision	0.6856	0.6782	0.6852	0.7004	0.7032	0.7362	0.7897	**0.8097***	+2.53%
	Recall	0.6068	0.6443	0.6519	0.6693	0.6547	0.7061	0.7647	**0.7859***	+2.77%
	AUC	0.7756	0.8445	0.8437	0.8461	0.8552	0.8809	0.9118	**0.9338***	+2.41%
Electronics	ACC	0.6313	0.6791	0.6598	0.6651	0.7002	0.7257	0.7339	**0.7413***	+1.01%
	Precision	0.6485	0.7089	0.7004	0.6812	0.7188	0.7463	**0.7614**	0.7570	-0.58%
	Recall	0.3846	0.4601	0.4554	0.4409	0.5182	0.5246	0.5982	**0.7304***	+22.10%
	AUC	0.8041	0.8568	0.8603	0.8417	0.8776	0.8851	0.9017	**0.9032***	+0.17%
Clothing, Shoes, and Jewelry	ACC	0.6145	0.6589	0.6441	0.6549	0.6688	0.6890	0.703	**0.7417***	+5.50%
	Precision	0.6586	0.6795	0.6607	0.6883	0.6998	0.7281	0.7365	**0.7408***	+0.58%
	Recall	0.4862	0.5641	0.5598	0.5699	0.5997	0.6015	0.6493	**0.7049***	+8.56%
	AUC	0.7563	0.8113	0.7998	0.8079	0.8209	0.8507	0.8544	**0.8808***	+3.09%

* We conduct a two-sided significant test between the proposed IRGNN and the strongest baseline, where * means the p-value is smaller than 0.05.

6) *HetGNN:* Heterogeneous GNN (HetGNN) [45] is the state-of-the-art GNN-based model designed for heterogeneous graphs with recurrent neural networks and the attention mechanism.
7) *LVAE:* LVAEs is the state-of-the-art model for predicting relationships between products [15]. LVAE is a generative deep learning model that links two VAEs using a connector neural network.

As the aforementioned models all assume that only one single relationship exists between items except RGCN, we replace the *Softmax* function before the output by a *Sigmoid* function to allow them to predict multiple relationships at the same time.

D. Evaluation Metrics

In discovering item relationships, the model should not only predict the existing relationships for a given pair of items but also the direction of the link. Given an ordered pair of items (each order represents one direction), we compute the area under the ROC curve (AUC), precision, recall, and accuracy (ACC) on the test set [59]. Note that we measure AUC, precision, and recall separately for each type of relationship and report the weighted average value. We take label imbalance into account so that the results are weighted by support (the number of true instances for each type of relationship). For the ACC score, all four types of relationships must be correctly predicted simultaneously.

E. Overall Performance (RQ1)

Table II shows the overall performance on Amazon data. Bold numbers are the best results. From the table, we can observe that LR has the lowest accuracy, indicating that the expressive power of the logistic model is insufficient to capture complex relationships among items. Sceptre improves LR by learning topic models to discover topics from the reviews. It not only learns the relationship between items but also the direction as well. However, LR and Sceptre only consider the information from direct neighbors. The connectivity of the neighbors a few hops away also contains rich semantics that could be used for discovering item relationships.

PME and PMSC outperform LR and Sceptre since they consider 2-hop connections in the item graph. PME models 1-hop and 2-hop proximities with metric learning approach. PMSC incorporates 2-hop path constraints by maximizing the co-occurrence patterns of the type of edges. However, both PME and PMSC require manually setting the dependencies in the graph and have limited expressive power. RGCN and HetGNN generally have better performance than PME and PMSC. It may be because RGCN and HetGNN can automatically model multihop relationships using GNNs. In particular, HetGNN aggregates different types of information with recurrent neural networks and the attention mechanism.

LVAE consists of two VAEs with a connector neural network, designed specifically to discriminate item relationships (substitutes and complements). LVAE is the strongest baseline since the pair of VAEs can capture meaningful item features explaining the relationship between items.

IRGNN consistently yields the best performance on all five data sets, even on the highly sparse data set such as Clothing, Shoes, and Jewelry. In particular, IRGNN improves over the strongest baseline LVAE by 3% in ACC, 9.7% in recall, and 1.9% in AUC on average, and the average precision is almost the same. We attribute this improvement due to IRGNN incorporating multihop relationships and better utilizing the edge relational features along the path. The edge relational features contain collaborative information such as substitute or complement and can be captured by the designed edge relational network in IRGNN.

In Table III, path-based methods, such as PME and PMSC, do not generalize well and fail to effectively identify item relationships. This might suggest that the four relationships

TABLE III
EXPERIMENTAL RESULTS ON TAOBAO DATA

	LR	Sceptre	PME	PMSC	RGCN	HetGNN	LVAE	**IRGNN**	imp%
ACC	0.7791	0.7837	0.7336	0.7729	0.8277	0.8446	0.8496	**0.859***	+1.11%
Precision	0.8152	0.8244	0.7008	0.7982	0.8608	0.8808	0.8803	**0.914***	+3.77%
Recall	0.7336	0.7216	0.8232	0.7310	0.7815	0.8012	0.8230	**0.8247***	+0.18%
AUC	0.8516	0.8530	0.8196	0.8382	0.8826	0.9038	0.9139	**0.9248***	+1.19%

* We conduct a two-sided significant test between the proposed IRGNN and the strongest baseline, where * means the p-value is smaller than 0.05.

in Taobao data such as coclick/copurchase/favor-both share similar semantic meanings and is a more difficult scenario to predict the item relationships and differentiate between them. Hence, the simple path constraints as in PME or PMSC may not be able to characterize the underlying item dependencies. Yet, IRGNN achieves significant improvements over all baselines under various metrics, showing its high effectiveness and generalization ability to complex product relationships and scenarios. Specifically, its relative improvement over the strongest baseline with respect to Precision is 3.77%.

F. Data Sparsity Problem (RQ2)

As discussed in Section III, our proposed model is capable of naturally utilizing multihop relationships of products, which can provide more information when data are sparse. In this section, we further evaluate the effectiveness of IRGNN as the degree of data sparsity increases on Amazon data. We decrease the ratio of the training set to increase the data sparsity and generate four versions of data sets. The data split ratios are varied from 8/1/1 (i.e., the ratio for training, validation, and test sets, respectively) to 4/3/3, 6/7/7, and 1/2/2.

The comparison between our proposed model and the strongest baseline LVAE on these four versions of the data sets is presented in Fig. 3. We observe that IRGNN consistently outperforms the strongest baseline in terms of accuracy on all sparse data sets. Moreover, the sparser the data set is, the larger improvement can be achieved by IRGNN upon LVAE. For example, from data set 8/1/1 to the sparsest data set 1/2/2, improvement of IRGNN upon LVAE increases from 2.8% to 3.2% in Video Games, from 2.5% to 4.5% in Musical Instruments, from 3.2% to 4.6% in Movies and TV, from 1% to 2.3% in Electronics, and from 5.5% to 8.4% in Clothing, Shoes, and Jewelry. This result demonstrates that IRGNN is superior to other methods in sparse data sets and shows that utilizing multihop relationships of products is effective to alleviate the data sparsity problem for item relationship prediction.

G. Hyperparameter Study (RQ3)

In this section, we study the impact of the hyperparameters on the performance of IRGNN.

1) Number of Message Propagation Iterations: As discussed in Section IV-F, multihop relationships of products can provide additional information in sparse data sets and hence benefit the prediction of item relationships. Naturally, the next question would be how many hops are most helpful for item relationship prediction? Hence, we study the impact of the number of message propagation iterations, which corresponds to the maximum number of hops of the connections. In Table IV, we present the performance in terms of ACC and AUC of IRGNN by varying the number of message propagation iterations (taking Video Games, Musical Instruments, and Movies and TV as examples due to limited places, similar observations can be obtained on the other two data sets). We observe that generally, a large number of iterations are more effective to predict item relationships. For example, IRGNN achieves the best performance when the number is 5 at Video Games and Movies and TV, and the best results when the number is 4 at Musical Instruments. This result demonstrates that the proposed model benefits from a large number of message propagation iterations.

TABLE IV
IRGNN WITH DIFFERENT NUMBERS OF MESSAGE PROPAGATION ITERATIONS

	Video Games		Musical Instruments		Movies & TV	
	ACC	AUC	ACC	AUC	ACC	AUC
IRGNN-2	0.7984	0.9442	0.7660	0.9268	0.8185	0.9212
IRGNN-3	0.8188	0.9568	0.7789	0.9307	0.8264	0.9261
IRGNN-4	0.8147	0.9560	**0.8129**	**0.9505**	0.8164	0.9200
IRGNN-5	**0.8403**	**0.9631**	0.7912	0.9477	**0.8322**	**0.9338**

2) Dimension of the Node Embeddings: Another impactful hyperparameter is the dimension of the node embeddings. We evaluate the proposed model with the dimension varying from 8 to 16 and 32 and present the results in Table V. We observe that IRGNN achieves the best performance when the dimension of the node embeddings is 16 for all data sets (results on the other two data sets are similar and omitted for simplicity). For example, IRGNN achieves the best accuracy of 0.8403 when the dimension is 16 on the Video Games data set, which is superior to 0.7836 when the dimension is 8 and slightly better than 0.8398 when the dimension is 32. On the one hand, IRGNN suffers from a small dimension (e.g., 8) due to its limited fitting capability. On the other hand, it is also worth noting that a large dimension may cause overfitting and degrade the performance.

H. Ablation Study (RQ4)

1) Shortcut Connections: As discussed in Section III, the goal of shortcut connections is to simplify the learning of multihop relationships. The useful information from the previous message propagation iteration could be directly utilized through this shortcut connection. In this section, we further

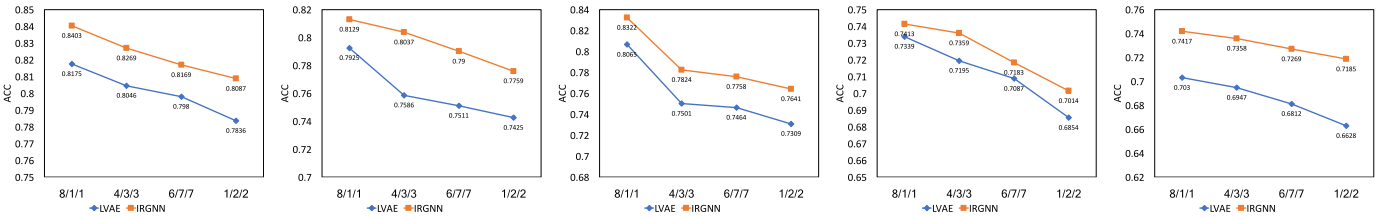


Fig. 3. Comparison between IRGNN and the strongest baseline in terms of ACC with different data sparsities. (a) Video games. (b) Musical instruments. (c) Movies and TV. (d) Electronics. (e) Clothing.

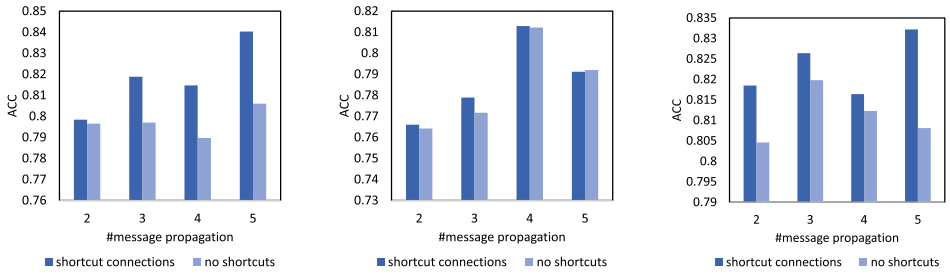


Fig. 4. IRGNN with and without shortcut connections. (a) Video games. (b) Musical instruments. (c) Movies and TV.

TABLE V
IRGNN WITH DIFFERENT NODE EMBEDDING DIMENSIONS

	Video Games		Musical Instruments		Movies & TV	
	ACC	AUC	ACC	AUC	ACC	AUC
IRGNN-8	0.7836	0.9206	0.8067	0.9432	0.7762	0.8891
IRGNN-16	**0.8403**	**0.9631**	**0.8129**	**0.9505**	**0.8322**	**0.9338**
IRGNN-32	0.8398	0.9537	0.8111	0.9469	0.8284	0.9280

TABLE VI
VARIANTS OF IRGNN

	Video Games		Musical Instruments		Movies & TV	
	ACC	AUC	ACC	AUC	ACC	AUC
IRGNN-SIMPLE-E	0.8378	0.9486	0.8089	0.9412	0.8288	0.9284
IRGNN-LIN-E	0.8386	0.9477	0.8115	0.9429	0.8256	0.9248
IRGNN-GRU	0.8384	0.9499	0.8078	0.9131	0.8157	0.9186
IRGNN-LIN-D	0.8164	0.9307	0.8070	0.9435	0.8125	0.9078
IRGNN	**0.8403**	**0.9631**	**0.8129**	**0.9505**	**0.8322**	**0.9338**

evaluate the effectiveness of the design of the shortcut connections in IRGNN. We vary the number of message propagation iterations and report the accuracy for the proposed model with and without shortcut connections in Fig. 4. We observe that the shortcut connection is truly helpful to improve the performance. For example, IRGNN with the shortcut connections outperforms the variant without the shortcut connections on Video Games for all numbers of message propagation, where the largest improvement is obtained when the number of message propagation is 5. A similar observation is also obtained on the other data sets except for Musical Instruments where IRGNN with and without the shortcut connections have a close performance. A possible reason behind this result is that the graph of Musical Instruments has many circles of length 4, and the message propagation is trapped by the circles, so that increasing the number of iterations will not improve the performance.

2) Other Variants of IRGNN: We also test several variants of IRGNN to validate the design of our edge relational network. Each time, we replace one key component and compare the result with the full-fledged IRGNN, where the results are presented in Table VI. Experimental results demonstrate that all the components working together yield the best performance.

a) Node embeddings in message propagation: We first evaluate the performance of IRGNN with a simple edge relational network that removes node embeddings and only takes as input the edge relational vector (IRGNN-SIMPLE-E), as adopted in [46], and namely, the edge relational network is reduced to

$$g\left(h_w^{(l)}, e_{vw}, h_v^{(l)}\right) = \sigma\left(W_{g'}^{(l)} e_{vw}\right).$$

In this setting, the transformation matrix g only depends on the edge relational vectors so that edges with the same edge relational vectors have the same transformation matrix. The results in Table VI indicate that the performance of IRGNN-SIMPLE-E is downgraded compared to IRGNN, which shows the effectiveness of node embeddings in message propagation.

b) Second-order feature interactions: To validate the usage of the elementwise product in the edge relational network, we compare IRGNN with the edge relational network that uses the linear concatenation of the source and destination node embeddings (IRGNN-LIN-E)

$$g\left(h_w^{(l)}, e_{vw}, h_v^{(l)}\right) = \sigma\left(W_{g''}^{(l)} \cdot \left[e_{vw} || h_w^{(l)} || h_v^{(l)}\right]\right).$$

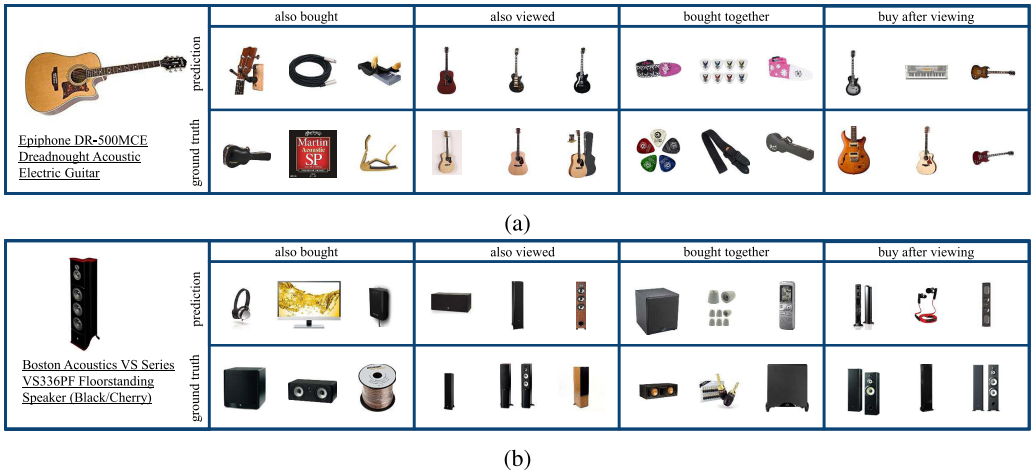


Fig. 5. Illustration of item relationship prediction. (a) Musical instruments. (b) Electronics.

As shown in Table VI, by using the elementwise product, ACC increases by 0.2%, 0.2%, and 0.8% for Video Games, Musical Instruments, and Movies and TV, respectively. This illustrates the importance of the second-order feature interactions.

c) GRU to replace shortcut connections: For the message update function, we propose to use two fully connected layers with a shortcut connection to facilitate the training with multihop relationships, which is more efficient than GRU and is capable of incorporating the message from the previous layer. We replace our designed f_u in (7) by a GRU as introduced in [30] (IRGNN-GRU) and evaluate the performance of it. Experimental results demonstrated the effectiveness of the shortcut connection, as IRGNN outperforms IRGNN-GRU on all the three data sets. It may be because the IRGNN-GRU overfits the training data.

d) Outer product: Although the simple concatenation is a widely adopted method for multirelationship prediction [15], we observe that the outer product may be more feasible for the item multirelationship prediction task. We replace the outer product in IRGNN with a simple concatenation of the source and destination node embeddings and propose IRGNN-LIN-D, which is equivalent to LR

$$\hat{e}_{vw}^{(i)} = \mu\left(W_{o'}^{(i)} \cdot \left[h_v^{(L)} || h_w^{(L)}\right]\right)$$

where μ is the sigmoid function. Table VI demonstrates that the outer product enhances the performance of the multi-item relationship prediction, with ACC increasing by 2.9%, 0.7%, and 2.4%, AUC increasing by 3.5%, 0.7%, and 2.9% on Video Games, Musical Instruments, and Movies and TV, respectively.

3) Neighborhood Sampling: As discussed in Section III-C2, without neighborhood sampling, the memory and expected runtime of a single batch is unpredictable and in the worst case is $O(n)$, where n is the number of nodes in the product graph. Therefore, we study the effectiveness and efficiency of our proposed IRGNN with and without neighborhood sampling on Video Games, as shown in Table VII. The runtime in Table VII records the average running time of a message propagation step for one node. The results suggest that neighborhood sampling could help IRGNN achieve higher performance with lower space and time complexity since message propagation without neighborhood sampling has to aggregate messages from all neighbors, which is less efficient and may involve noisy neighbors.

TABLE VII
IRGNN WITH AND WITHOUT NEIGHBORHOOD SAMPLING

w/ Neighborhood Sampling			w/o Neighborhood Sampling		
ACC	AUC	Runtime (μs)	ACC	AUC	Runtime (μs)
0.8403	0.9631	2.07	0.8087	0.9306	4.27

TABLE VIII
IRGNN WITH DIFFERENT RATIOS OF POSITIVE TO NEGATIVE SAMPLES

	Video Games		Musical Instruments		Movies & TV	
pos:neg	ACC	AUC	ACC	AUC	ACC	AUC
1:1	0.8403	**0.9631**	0.8129	0.9505	0.8322	0.9338
1:3	0.836	0.9589	0.8158	0.9324	**0.8326**	0.933
1:5	**0.8422**	0.9614	**0.8214**	**0.9533**	0.8255	**0.9372**

4) Negative Sampling: Directly treating all unobserved edges as negative samples is extremely computationally expensive, as the number of the unobserved edges is huge and power to the number of nodes [19]. To this end, we uniformly sample negative samples as in [23] and study the effect of the ratio of positive to negative samples. As shown in Table VIII, IRGNN is generally not sensitive to the ratio of positive to negative samples—different settings yield similar results. The performance is getting slightly better when adding more negative examples. Yet, more negative samples require more time and space for training the model, and we use the 1:1 balanced setting in our experiments.

I. Case Study (RQ5)

1) Item Relationship-Based Recommendation: To perform recommendation according to the learned item relationships, we randomly select a query item. We rank the remaining

TABLE IX
EXAMPLES OF PREDICTIONS ON MULTIHOP CONNECTIONS (TWO TYPES OF RELATIONSHIPS: SUB/COMP)

		Musical Instruments			Electronics	
Multi-hop paths	Ratio	Positive cases	Negative cases	Ratio	Positive cases	Negative cases
(sub, sub, sub) $\Rightarrow$ sub	91.9%	Alesis Q25 Keyboard Controller Akai LPK25 Keyboard Controller Alesis Q49 Keyboard Controller Akai MPK Keyboard Controller	Cecilio 4/4 CVN-200 Violin Cecilio CVN-300 Violin Mendini 4/4 MV300 Violin Crescent 4/4 Student Violin Starter Kit	88.5%	Time Warner Cable Remote Control OARN08G 8 Device Remote Universal Remote Control URC7140 Logitech Harmony 300 Remote Control	3M Notebook Screen Cleaning Wipes iCloth touchscreen cleaning wipes Fellowes Screen Cleaning Wipes Screen Cleaner Kit
(comp, comp, comp) $\Rightarrow$ comp	87.3%	Eric Clapton - Acoustic Classics (DVD) Eric Clapton - The Early Years (DVD) Eric Clapton - The Solo Years DVD B.B. King: Blues Master	Hercules GS414B A/G Guitar Stand Hercules GSB001 Carry Bag On-Stage Folding Orchestral Music Stand On-Stage Flip-It A-Frame Guitar Stand	86.9%	HP Remote IR receiver Windows Media Center Extender URC WR7 Universal Remote Control AmazonBasics High-Speed HDMI Cable	Logitech Harmony 600 Universal Remote HP Remote IR receiver Windows Media Center Extender URC WR7 Universal Remote Control
(comp, comp, sub) $\Rightarrow$ comp	78.5%	On Stage XCG4 Tubular Guitar Stand Black Single-handed Guitar Capo Snark SN-5 Tuner Fender FT-004 Chromatic Clip-On Tuner	Vandoren CG100B Clarinet Cork Grease SuperSlick HSWAB Clarinet Hanky Swab Woodwind Cork Grease Venture 200 Tube Cork Grease	74.2%	9 cell battery for Dell D620 Hard Drive Caddy for Laptop Dell D620 Internal Generic DVD-RW Burner For D620 Silm 8X DVDRW Drive Burner for D620	Apple MagSafe 2 Power Adapter MagSafe to MagSafe 2 Converter Apple MagSafe 2 Power Adapter (Open Box) New Apple Magsafe2 Ac Power Adapter

TABLE X
PREDICTIONS ON MULTIHOP CONNECTIONS (FOUR TYPES OF RELATIONSHIPS: AB/AV/BT/BV)

	Multi-hop paths	Ratio
Musical Instruments	(AB, BT, AB) $\Rightarrow$ AB	85.54%
	(AV, AV, AV) $\Rightarrow$ AV	81.33%
	(AB, AB, AB) $\Rightarrow$ AB	80.61%
Electronics	(AB, BT, AB) $\Rightarrow$ AB	84.86%
	(AB, AB, AB) $\Rightarrow$ AB	82.73%
	(AV, BV, AV) $\Rightarrow$ AV	82.55%

items by the predicted score for each type of relationship with the query item. Illustrations of the top-3 recommendation are shown in Fig. 5. For the convenience of visualization, we use the category of Musical Instruments and Electronics. For the ground-truth relationships, since the data are sparse and are hard to find an item in the test set that contains all four relationships, we need to densify the data set. We perform a content-based technique [60] and create pseudoitems by grouping the ten-nearest neighbors to one pseudoitem according to the item features. Ground-truth relationships of the pseudoitems are presented in the second row.

Fig. 5 shows that compared to the ground truth, IRGNN can perform reasonable recommendations according to multiple complicated relationships. For example, IRGNN is capable to learn the pattern that users *also viewed* other electronic guitars after viewing the queried guitar and *bought* guitar picks and shoulder straps *together* with the guitars. Such results may also be useful for understanding item relationships and user behavior prediction.

2) Multihop Prediction: To investigate whether our proposed IRGNN is capable of learning meaningful knowledge from multihop connections, we use IRGNN to predict item relationships where items are connected by a 3-hop path. We count predictions made on three most frequent 3-hop paths as an illustration. To start with, we consider edges with single relationships (either complements or substitutes) for easier understanding. We refer to AB and BT as complements and AV and BV as substitutes. Table IX computes the ratio of the number of the given patterns to all the predictions made by IRGNN. We also show examples of positive and negative cases. Four products are connected sequentially by the given relationships and we let that IRGNN predicts the item relationship between the first and the last item. Positive cases represent the cases that the item relationship predicted by IRGNN is in accordance with the given pattern such as [(sub, sub, sub) $\Rightarrow$ sub] on the left, whereas negative cases represent that IRGNN predicts the opposite (e.g.[(sub, sub, sub) $\Rightarrow$ comp/unrelated]). Results show that IRGNN can discover useful dependencies of substitutes and complements among 3-hop paths. For example, IRGNN learns that if $i_0 \xrightarrow{\text{sub}} i_1 \xrightarrow{\text{sub}} i_2 \xrightarrow{\text{sub}} i_3$, then likely $i_0 \xrightarrow{\text{sub}} i_3$, which matches our intuition. As shown in Table IX, in Musical Instruments, i_0, i_1, i_2, and i_3 are four keyboard controllers but with different brands, and then i_0 and i_3 are also substitutable. Negative cases suggest that apart from the edge relational information, IRGNN also considers the item content information to make predictions.

We further display the multihop dependencies with multiple relationships learned by IRGNN in Table X. We compute the ratio of the frequency of the given multihop pattern to all the predictions made by IRGNN. Table X presents the three most frequent patterns and reveals that IRGNN can discover multihop patterns such as [(AB, BT, AB) $\Rightarrow$ AB]. In sum, IRGNN is able to preserve useful path dependencies and thereby improves the performance of item relationship prediction.

V. CONCLUSION AND FUTURE WORK

In this work, we propose a GNN-based framework, IRGNN, with multihop relationships to discover item multirelationships. Rather than solely relying on the item content information, IRGNN automatically learns topological features and relational dependencies in multihop neighborhoods to improve the quality of item relationship prediction. The edge relational network is carefully designed to impart both node and edge features to the message propagation process. Moreover, we study a more generalized setting—multigraph—for discovering item relationships. Extensive experiments on large real-world item graphs demonstrate the rationality and effectiveness of IRGNN.

This work exploits multihop connections in inferring item relationships. Multihop connections offer the potential of preserving complex relational dependencies that can help identify item relationships. This is the first work that emphasizes on edge relational features for the item relationship prediction. We hope that it can provide insights on how to better model the complex item relationships in real-world e-commerce scenarios.

In practice, item relationships may change dynamically and evolve over time due to daily transactions, which may not be handled well by our current model due to the assumption of static training data [61]–[63]. An online variant of IRGNN on how to efficiently update the model can be investigated in the future. The temporal evolutionary patterns can also be incorporated to improve the representation quality of items. In addition, as item relationships may vary for individuals, we plan to personalize the message propagation step and thereby generate personalized recommendation in our continuing work.

REFERENCES

[1] B. J. Mirza, B. J. Keller, and N. Ramakrishnan, "Studying recommendation algorithms by graph analysis," *J. Intell. Inf. Syst.*, vol. 20, no. 2, pp. 131–160, 2003.

[2] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 165–174.

[3] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 311–319.

[4] X. Li and H. Chen, "Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach," *Decis. Support Syst.*, vol. 54, no. 2, pp. 880–890, Jan. 2013.

[5] W. Fan *et al.*, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 417–426.

[6] L. Wu, P. Sun, R. Hong, Y. Fu, X. Wang, and M. Wang, "SocialGCN: An efficient graph convolutional network based model for social recommendation," 2019, *arXiv:1811.02815*. [Online]. Available: https://arxiv.org/abs/1811.02815

[7] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1164–1177, May 2017.

[8] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 785–794.

[9] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 43–52.

[10] Z. Wang, Z. Jiang, Z. Ren, J. Tang, and D. Yin, "A path-constrained framework for discriminating substitutable and complementary products in E-commerce," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 619–627.

[11] Y. Zhang, H. Lu, W. Niu, and J. Caverlee, "Quality-aware neural complementary item recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 77–85.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[13] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[14] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2018, pp. 593–607.

[15] V. Rakesh, S. Wang, K. Shu, and H. Liu, "Linked variational Autoencoders for inferring substitutable and supplementary items," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 438–446.

[16] R. He, C. Packer, and J. McAuley, "Learning compatibility across categories for heterogeneous item recommendation," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 937–942.

[17] Q. Zhao, Y. Zhang, Y. Zhang, and D. Friedman, "Multi-product utility maximization for economic recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 435–443.

[18] T. Chen, H. Yin, G. Ye, Z. Huang, Y. Wang, and M. Wang, "Try this instead: Personalized and interpretable substitute recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 891–900.

[19] S. Zhang, H. Yin, Q. Wang, T. Chen, H. Chen, and Q. V. H. Nguyen, "Inferring substitutable products with deep network embedding," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4306–4312.

[20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903

[21] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[22] T. Derr, Y. Ma, and J. Tang, "Signed graph convolutional networks," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 929–934.

[23] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for Web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.

[24] A. Dutta and H. Sahbi, "Stochastic graphlet embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 8, pp. 2369–2382, Aug. 2019.

[25] W. Liu, Q. Liu, R. Tang, J. Chen, X. He, and P. A. Heng, "Personalized re-ranking with item relationships for E-commerce," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 925–934.

[26] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882

[27] K. Ding, J. Wang, J. Li, D. Li, and H. Liu, "Be more with less: Hypergraph attention networks for inductive text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 4927–4936.

[28] Y. Fan, J. C. Lam, and V. O. Li, "Multi-region ensemble convolutional neural network for facial expression recognition," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2018, pp. 84–94.

[29] C. Morris *et al.*, "Weisfeiler and Leman go neural: Higher-order graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4602–4609.

[30] X. Bresson and T. Laurent, "Residual gated graph ConvNets," 2017, *arXiv:1711.07553*. [Online]. Available: http://arxiv.org/abs/1711.07553

[31] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 555–563.

[32] P. Liu, L. Zhang, and J. A. Gulla, "Real-time social recommendation based on graph embedding and temporal context," *Int. J. Hum.-Comput. Stud.*, vol. 121, pp. 58–72, Jan. 2019.

[33] J. Wang, K. Ding, Z. Zhu, Y. Zhang, and J. Caverlee, "Key opinion leaders in recommendation systems: Opinion elicitation and diffusion," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 636–644.

[34] L. Grad-Gyenge, A. Kiss, and P. Filzmoser, "Graph embedding based recommendation techniques on the knowledge graph," in *Proc. Adjunct Publication 25th Conf. User Model., Adaptation Personalization*, Jul. 2017, pp. 354–359.

[35] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: A review of ontology-based recommender systems for E-learning," *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 21–48, Jun. 2018.

[36] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," 2019, *arXiv:1905.08108*. [Online]. Available: http://arxiv.org/abs/1905.08108

[37] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*. [Online]. Available: http://arxiv.org/abs/1706.02263

[38] J. Wang, K. Ding, L. Hong, H. Liu, and J. Caverlee, "Next-item recommendation with sequential hypergraphs," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1101–1110.

[39] D. J. Klein and M. Randić, "Resistance distance," *J. Math. Chem.*, vol. 12, no. 1, pp. 81–95, Dec. 1993.

[40] F. Aziz, R. C. Wilson, and E. R. Hancock, "Backtrackless walks on a graph," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 977–989, Jun. 2013.

[41] M. Zhang and Y. Chen, "Weisfeiler-lehman neural machine for link prediction," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 575–583.

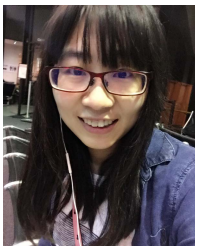
[42] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: Projected metric embedding on heterogeneous networks for link prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1177–1186.

[43] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5165–5175.

[44] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," 2017, *arXiv:1709.05584*. [Online]. Available: http://arxiv.org/abs/1709.05584
[45] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 793–803.
[46] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.
[47] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–14.
[48] Y. Qu *et al.*, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1149–1154.
[49] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 355–364.
[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[51] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[52] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: http://arxiv.org/abs/1409.1259
[53] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–20.
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[55] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," 2018, *arXiv:1812.08434*. [Online]. Available: http://arxiv.org/abs/1812.08434
[56] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[57] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 507–517.
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2014, pp. 1–15.
[59] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011, *arXiv:2010.16061*. [Online]. Available: https://arxiv.org/abs/2010.16061
[60] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, 1997.
[61] T. Chen, H. Yin, Q. V. H. Nguyen, W.-C. Peng, X. Li, and X. Zhou, "Sequence-aware factorization machines for temporal predictive analytics," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Apr. 2020, pp. 1405–1416.
[62] W. Wang, H. Yin, Z. Huang, Q. Wang, X. Du, and Q. V. H. Nguyen, "Streaming ranking based recommender systems," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 525–534.
[63] W. Liu, F. Liu, R. Tang, B. Liao, G. Chen, and P. A. Heng, "Balancing between accuracy and fairness for interactive recommendation with reinforcement learning," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2020, pp. 155–167.

Weiwen Liu received the B.S. degree in computer science from South China University and Technology, Guangzhou, China, in 2016, and the Ph.D. degree in computer science from The Chinese University of Hong Kong, Hong Kong, in 2020.

Her research interests include machine learning algorithms, information retrieval, and recommender systems.

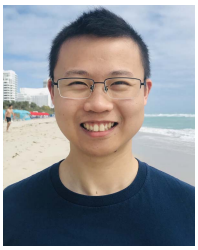


Yin Zhang received the B.S. degree in mathematics from Beijing Normal University, Beijing, China, in 2014, and the M.S. degree in statistics from the University of California at Davis, Davis, CA, USA, in 2016. She is currently pursuing the Ph.D. degree in computer science and engineering with Texas A&M University, College Station, TX, USA.

Her research interests include recommendation and machine learning.

Jianling Wang received the bachelor's degree in information engineering from The Chinese University of Hong Kong, Hong Kong, in 2015, and the master's degree in computer science from the Georgia Institute of Technology, Atlanta, GA, USA, in 2017. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA.

Her research interests generally lie in data mining, machine learning, and recommendation systems.



Yun He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA.

His interests include pretrained language models, information retrieval, and recommender systems.

James Caverlee (Member, IEEE) received the B.A. degree *(magna cum laude)* in economics from Duke University, Durham, NC, USA, in 1996, the M.S. degrees in engineering-economic systems and operations research and computer science from Stanford University, Stanford, CA, USA, in 2000 and 2001, respectively, and the Ph.D. degree in computer science from Georgia Tech, Atlanta, GA, USA, in 2007.

He is a Professor and Lynn '84 and Bill Crane '83 Faculty Fellow with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA.

Dr. Caverlee was a recipient of the NSF CAREER Award, the DARPA Young Faculty Award, and the AFOSR Young Investigator Award, as well as several teaching awards. He was the Conference General Co-Chair for WSDM 2020.

Patrick P. K. Chan (Senior Member, IEEE) received the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, in 2009.

He is currently an Associate Professor with the School of Computer Science and Engineering, and the person in charge of machine learning and the Cybernetics Research Laboratory, South China University of Technology, Guangzhou, China. He is also a part-time Lecturer with the Hyogo College of Medicine, Nishinomiya, Japan. His current research interests include pattern recognition, multiple classifier systems, biometric, computer security, deep learning, and reinforcement learning.

Dr. Chan was a member of the governing boards of the IEEE SMC Society from 2014 to 2016. He serves as the Organizing Committee Chair for several international conferences. He was also the Chairman of the IEEE SMCS Hong Kong Chapter 14–15. He is the Counselor of the IEEE Student Branch, South China University of Technology. He is an Associate Editor for international journals, including *Information Sciences* and the *International Journal of Machine Learning and Cybernetics*.

Daniel S. Yeung (Life Fellow, IEEE) received the Ph.D. degree in applied mathematics from Case Western Reserve University, Cleveland, OH, USA, in 1974.

He was an Assistant Professor of mathematics and computer science with the Rochester Institute of Technology, Rochester, NY, USA, as a Research Scientist with the General Electric Corporate Research Center, USA, and as a System Integration Engineer with TRW, USA. He was a Visiting Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, from 2008 to 2015. He was the Chairman of the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, and a Chair Professor from 1999 to 2006. His current research interests include neural-network sensitivity analysis, data mining, and big data analytic.

Dr. Yeung is the Past President of the IEEE Systems and the Man and Cybernetics Society. He is the Co-Editor-in-Chief of the *Springer International Journal on Machine Learning and Cybernetics*.

Pheng-Ann Heng (Senior Member, IEEE) received the B.Sc. degree in computer science from the National University of Singapore, Singapore, in 1985, the M.Sc. degree in computer science, the M.Art. degree in applied mathematics, and the Ph.D. degree in computer science from Indiana University, Bloomington, IN, USA, in 1987, 1988, and 1992, respectively.

He has served as the Department Chairman from 2014 to 2017 and as the Head for the Graduate Division from 2005 to 2008 and then again from 2011 to 2016. He has been serving as the Director for the Virtual Reality, Visualization, and Imaging Research Center, The Chinese University of Hong Kong (CUHK), since 1999. He has been serving as the Director of the Center for Human–Computer Interaction, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, since 2006. He has been appointed by the China Ministry of Education as a Cheung Kong Scholar Chair Professor in 2007. He is currently a Professor with the Department of Computer Science and Engineering, CUHK. His research interests include AI and VR for medical applications, surgical simulation, visualization, graphics, and human–computer interaction.