Contextualized Graph Attention Network for Recommendation with Item Knowledge Graph

Yong Liu, Susen Yang, Yonghui Xu, Chunyan Miao, Min Wu, and Juyong Zhang

Abstract—Graph neural networks (GNN) have recently been applied to exploit knowledge graph (KG) for recommendation. Existing GNN-based methods explicitly model the dependency between an entity and its local graph context in KG (*i.e.*, the set of its first-order neighbors), but may not be effective in capturing its non-local graph context (*i.e.*, the set of most related high-order neighbors). In this paper, we propose a novel recommendation framework, named Contextualized Graph Attention Network (CGAT), which can explicitly exploit both local and non-local graph context information of an entity in KG. More specifically, CGAT captures the local context information by a user-specific graph attention mechanism, considering a user's personalized preferences on entities. In addition, CGAT employs a biased random walk sampling process to extract the non-local context of an entity, and utilizes a Recurrent Neural Network (RNN) to model the dependency between the entity and its non-local contextual entities. To capture the user's personalized preferences on items, an item-specific attention mechanism is also developed to model the dependency between a target item and the contextual items extracted from the user's historical behaviors. We compared CGAT with state-of-the-art KG-based recommendation methods on real datasets, and the experimental results demonstrate the effectiveness of CGAT.

Index Terms—Recommendation systems, knowledge graph, graph neural networks.

1 Introduction

Nowdays, with the explosive growth of online information and contents, recommendation systems have played an increasingly important role in various scenarios, such as E-commerce websites and online social media platforms. The recommendation systems usually aim to provide the user a list of items that she may have interests. In the last decades, various techniques, *e.g.*, collaborative filtering [1] and deep learning [2], have been proposed to utilize the user behavior data to solve different recommendation problems. Despite many research efforts have been devoted to developing recommendation systems, most existing methods that only consider the user-item interaction information still suffer from the data sparsity and cold-start problems. To remedy these problems, various side information has been incorporated into recommendation systems, such as users' social networks [3] and review texts [4]. Among various types of side information, the item knowledge graph (KG) with rich relations between items has been shown to be effective in improving recommendation performances [5]. Essentially, KG is a heterogeneous network where nodes correspond to entities and edges correspond to relations. Several KGs, such as DBpedia[1] and Microsoft Satori[2], have been successfully applied in many scenarios, for example KG completion [6] and question answering [7]. Inspired by the success of these applications, some recent efforts have been devoted to utilizing KG to improve recommendation performances [8], [9], [10], [11], [12], [13], [14], [15], [16]. The main challenge of incorporating KG for recommendation is how to effectively exploit the relations between entities and the graph structure of KG.

In general, previous recommendation methods employing item KG can be roughly categorized into three groups: 1) regularization-based approaches [8], [14], 2) path-based approaches [9], [10], and 3) graph neural network (GNN)-based approaches [15], [16]. The regularization-based methods impose well-designed additive regularization loss terms to capture the KG structure. However, they only encode the KG relation in an implicit manner and can not explicitly consider the semantic relation information of KG into the recommendation model. The path-based methods focus on extracting the high-order connectivity information between entities along paths which are always manually designed or selected based on special criteria. These approaches may heavily rely on domain knowledge. Recently, the quick development of graph neural networks (GNN) [17] motivates the application of graph convolutional networks (GCN) [18] and graph attention networks (GAT) [19] in developing end-to-end recommender systems [15], [16], [20], [21], [22], which use GNN to automatically aggregate the context

- *Yong Liu is currently with Alibaba-NTU Singapore Joint Research Institute and Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore 639798. Email: stephenliu@ntu.edu.sg.*
- *Susen Yang is currently with Alibaba Group, Hangzhou, China 311121, and School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, China 230026. Email: susen.yss@alibaba-inc.com.*
- *Yonghui Xu is currently with Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan, China, 250101. Email: xu.yonghui@hotmail.com.*
- *Chunyan Miao is currently with School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. Email: ascymiao@ntu.edu.sg.*
- *Min Wu is currently with the Institute for Infocomm Research (I^2R), A*Star, Singapore 138632. Email: wumin@i2r.a-star.edu.sg.*
- *Juyong Zhang is currently with School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui, China 230026. Email: juyong@ustc.edu.cn.*
- *Corresponding authors: Min Wu, Juyong Zhang.*

Manuscript received xxx, 2020; revised xxx, 2020.

1. http://wiki.dbpedia.org/
2. https://searchengineland.com/library/bing/bing-satori

information from structural neighbors of an entity in KG.

Generally speaking, the graph context of an entity in the item KG includes local graph context which is the first-order neighbors and non-local graph context that denotes the set of most related high-order neighbors. The GNN-based knowledge-aware recommendation methods [15], [16], [21] are developed to capture both the structure and semantic information of KG by aggregating these context. However, these methods may still have some deficiencies, and thus can not well solve the following challenges.

- (**Challenge 1**) Most GNN-based methods only consider the items and entities in KG and lack of modeling user-specific preferences on entities, when aggregating the local graph context of an entity in KG. As shown in Figure 1, both users have interactions with the item i_2. However, they prefer i_2 may due to different reasons. For example, u_1 prefers i_2 because of the attribute entity e_1 of i_2 in KG, while u_2 pays more attentions to its attribute entity e_3. The methods that ignore this situation are insufficient to model users' personalized preferences. Thus, the first challenge is: *how to capture a user's personalized preferences on entities when aggregating neighboring information in KG?*
- (**Challenge 2**) The non-local graph context of an entity in KG is not effectively captured in existing GNN-based recommendation methods. The knowledge graphs are always incomplete and some important connections between entities may be missing. Moreover, some items may have very few neighbors in KG, thus some important entities may not be directly connected to them. For example, in Figure 1, the item i_4 has only one entity e_3 linked with it, thus the aggregation of local context information for the entity e_3 is not enough to represent i_4. Moreover, we can also observe that entity e_1 is connected with i_4 along many multi-hop paths (*e.g.*, $e_1 \rightarrow i_1 \rightarrow e_3 \rightarrow i_4$ and $e_1 \rightarrow i_3 \rightarrow e_3 \rightarrow i_4$), which demonstrates the importance of e_1 to i_4. Exiting GNN-based methods [15], [16], [21] address this limitation by feature propagation layer by layer. However, this kind of layer by layer propagation may weaken the effects of farther connected entities or even bring noise information. Thus, the second challenge is: *how to sample and aggregate the non-local graph context that is strongly related to representation learning for each item in KG?*
- (**Challenge 3**) In practice, the user's historical items usually play an important role in predicting her preferences on candidate items [23]. Intuitively, for different target items, the historical items of a user may have different contributions in predicting the user preferences. Most previous methods learn the user representation by directly aggregating her historical item information, but ignore the importance of different historical items. Therefore, the third challenge is: *how to incorporate different importance of the user's historical items to the target item when aggregating the historical item representations for learning the user preferences?*

To solve these challenges, we propose a novel recommendation framework, named Contextualized Graph Attention Network (CGAT), which explicitly exploits both the local and non-local context of an entity in KG, as well as the item context extracted from users' historical data. The contributions made in this paper are as follows:

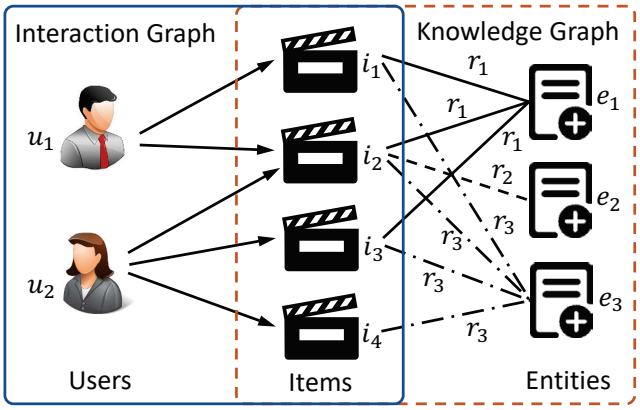


Fig. 1. A simple example showing the user-item interactions and the item knowledge graph.

- We propose a user-specific graph attention mechanism to aggregate the local context information in KG for recommendation, based on the intuition that different users may have different preferences on the same entity in KG.
- We propose to explicitly exploit the non-local context information in KG. More specifically, we develop a biased random walk sampling process to extract the non-local context of an entity, and employ a recurrent neural network (RNN) to model the dependency between the entity and its non-local context in KG.
- We develop an item-specific attention mechanism that exploits the context information from a user's historical behavior data to model her preferences on items.
- We perform extensive experiments on real datasets to demonstrate the effectiveness of CGAT. Experimental results indicate that CGAT usually outperforms state-of-the-art KG-based recommendation methods.

The remaining parts of this paper are organized as follows. Section 2 reviews the most relevant existing work about graph neural networks and KG-based recommendation. Section 3 introduces the formulation of the recommendation problem studied in this work. Next, Section 4 describes the details of the proposed recommendation model. In Section 5, we summarize the experimental results. Finally, in Section 6, we conclude this work and discuss potential directions for future work.

2 Related Work

In this section, we review existing work about knowledge graph-based recommendation and graph neural networks.

2.1 Knowledge Graph-based Recommendation

Most KG-based recommendation methods mainly aim to incorporate item KG to improve performances of recommendation. These methods can be categorized into three main groups: regularization-based methods, path-based methods, and GNN-based methods.

The regularization-based methods exploit the KG structure by imposing regularization terms into the loss function

to capture the KG structure and learn entity embedding. For example, the collaborative knowledge base embedding method [8] derives semantic entity representations from item KG, which is used to enhance collaborative filtering based recommendation. Considering the incomplete nature of KG, transfer learning is used to jointly train the item recommendation and KG completion tasks by regularizing or sharing entity and item embeddings if they refer to the same thing [24]. However, in this method, the entity relations are ignored in transferring knowledge from KG. To solve this problem, [14] aligns the preferences of users on items with the relations in KG, as well as aligning items with entities. Moreover, [25] proposes a cross&compress unit to approximated the high-order feature interactions between items and entities. The regularization-based methods are highly flexible. However, they lack an explicit modeling of the semantic relations in KG for recommendations.

The path-based methods exploit various connection patterns between entities to provide additional guidance for recommendations. For example, in [26], an attribute-rich heterogeneous information network (HIN) is built to enhance the recommendation quality. The meta-path based preference matrix between user and item in KG is estimated and decomposed by a Bayesian ranking model [27]. This method is then extended to learn personalized preferences by clustering users based on their interests [28]. In order to accurately capture semantic relations among users, the weighted graph and weighted meta-path concepts are proposed to more subtly reveal object relations through distinguishing link attribute values [29]. In [9], matrix factorization is used to generate latent features for users and items based on similarities generated by each meta-graph, and a factorization machine is applied to assemble different meta-graph based features. Moreover, the HIN embedding methods have also been studied for recommendation. In [30], the meta-path based random walk strategy is used to construct the heterogeneous neighborhood of a node, and then a heterogenous skip-gram model is utilized to learn the node embedding. Similarly, [31] employs a meta-path based sampling strategy to generate node sequence for HIN embedding. [10] employs a priority based sampling technique to select the high-quality meta-paths in HIN and uses co-attention mechanism to enhance the learning of the user, item, and meta-path context representations. To address the limitation of manually designed meta-paths, different selection rules or propagation methods have been proposed [12]. For example, in [32], the length condition is used to extract paths and then a batch of RNN is applied to aggregate the path information. Besides the length, multi-hop relational paths can also be inducted based on item associations [33]. The path-based methods heavily on manually designed meta-paths/meta-graphs, which rely on domain knowledge.

Recently, GNN-based methods [17] have been applied to develop the end-to-end KG-based recommender systems. For example, the knowledge graph convolutional network (KGCN) [21] employs non-spectral GCN to learn the structural information and semantic information of KG for recommendation. In [15], the KGNN-LS (*i.e.*, Knowledge-aware Graph Neural Networks with Label Smoothness regularization) model employs a trainable function that calculates the relation weights for each user to transfer the KG into a user-specific weighted graph, and then applies GCN on this graph to learn item embedding. In [16], the graph attention mechanism is adopted to aggregate and propagate local neighborhood information of an entity, without considering users' personalized preferences on entities. On summary, these GNN-based methods implicitly aggregate the high-order neighborhood information via layer by layer propagation, instead of explicitly modeling the dependency between an entity and its high-order neighbors.

In the literature, there exist some other applications of item KG. For example, in [34], a path recurrent network is proposed to learn representations for paths extracted in the item KG, considering the semantics of entities and relations. In [35], an explainable interaction driven user modeling algorithm is developed to exploit the item knowledge graph to build an explainable sequential recommender system. In [36], the temporal meta-path guided explainable recommendation model is proposed to capture the user-item evolutions on dynamic knowledge graph for explainable recommendation, based on attention mechanisms. Reinforcement learning has also been employed to perform reasoning on item KG in recent studies [37], [38], which aim to provide actual paths in KG to explain the recommendation results. Moreover, [39] proposes a novel negative sampling method called knowledge graph policy network that exploits the knowledge signal in KG to choose potential negative items for training accurate recommender systems.

2.2 Graph Neural Networks

Existing GNN methods can be categorized into three main groups [40]: recurrent graph neural networks, graph convolutional neural networks, and graph autoencoders.

Recurrent graph neural networks [41], [42], [43] are the pioneer GNN methods. In these methods, the representation of a node is learned by propagating its neighborhood information in an iterative manner until convergence. For example, in [41], the node's states are updated by exchanging its neighborhood information recurrently before satisfying the convergence criterion. To improve the training efficiency of [41], the Graph Echo State Network (GraphESN) [42] generalizes the echo state network to graph domains. Moreover, the Gate Graph Neural Networks (GGNN) [43] employs a gated recurrent unit as the recurrent function, which no longer needs parameter constraints to ensure convergence.

Graph convolutional neural networks (GCNs) generalize the convolution operation from grid data to graph data. In the literature, GCNs fall into two categories: spectral-based methods and spatial-based methods. The spectral-based methods [18], [44], [45] define convolution from the perspective of graph signal processing. For example, [44] employs the eigen decomposition of the graph Laplacian and defines the convolution in the Fourier domain. To reduce computational complexity, [45] approximates the convolutional filters by Chebyshey polynomials of the diagonal matrix of eigenvalues. A more simple convolutional architecture [18] can be inducted by the first-order approximation of spectral convolutions on graphs. The spatial-based methods [19], [46], [47], [48], [49], [50], [51] directly operate on the graph and propagate node information along

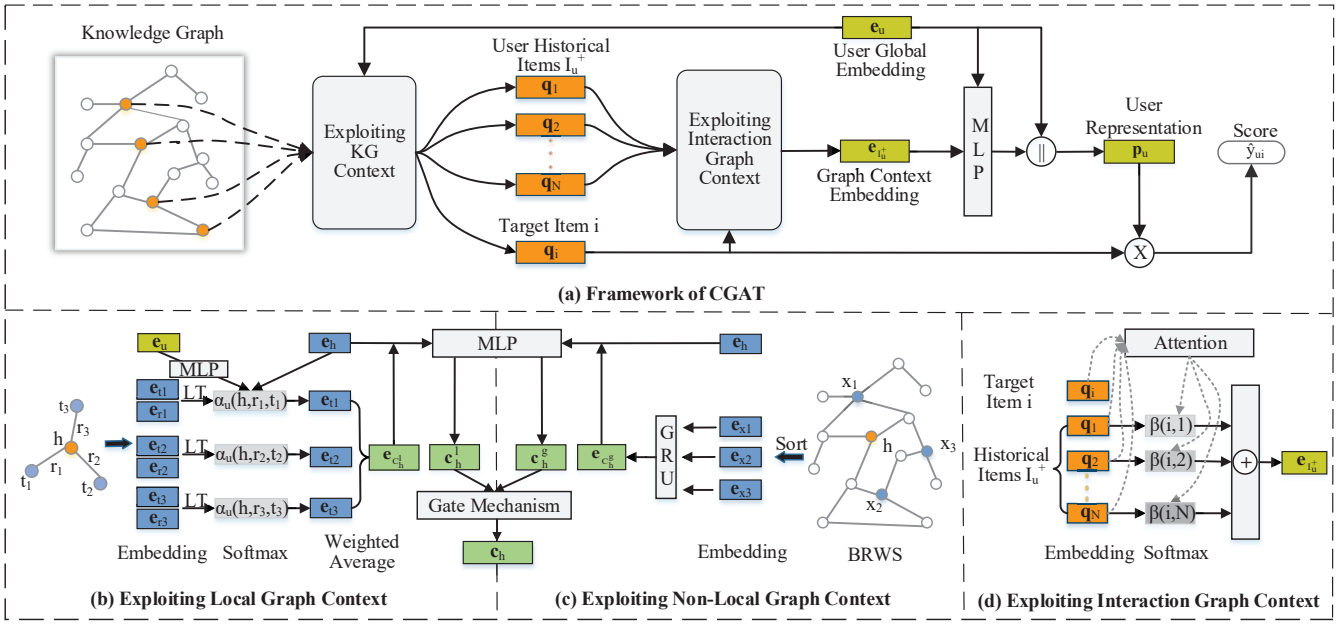


Fig. 2. (a) The framework of the CGAT. From left to right, it exploits the KG context and interaction graph context to predict a user's preference score on a candidate item; (b) Exploiting local graph context by applying a user-specific graph attention mechanism to KG; (c) Exploiting non-local graph context by biased random walk based sampling (BRWS) and GRU module; (d) Exploiting interaction graph context by applying an item-specific attention mechanism to the user's historical items.

edges. [46] proposes the first spatial-based method, which directly sums up a node's neighborhood information and applies residual connection to memorize information over each layer. [47] treats graph convolutions as a diffusion process and assumes there is a certain transition probability from a node to its neighbor. As the number of neighbors of nodes varies, [48] samples a fixed number of neighbors for each node and utilizes three different aggregators (*e.g.*, Long Short-Term Memory) to aggregate the neighbors' feature information. In [19], GAT assumes the contributions of neighbor nodes to the central node are different and performs multi-head attention mechanism to calculate the importance of different neighbors. Moreover, [49] employs a self-attention mechanism to compute an additional attention score for each attention head. [52] proposes a novel embedding method for bipartite graph, which employs a self-attention-based sequence modeling paradigm to capture the inter-class and intra-class message passing between explicit relations and implicit higher-order relations, respectively. To sufficiently describe the graph knowledge, a dual graph convolutional network [50] is proposed to consider both the local and global graph consistency by two parallel graph convolutional layers. [51] identifies the graph structures that cannot be distinguished by popular GNN methods, and proposes the graph isomorphism network which uses a learnable parameter to adjust the weight of the central node.

The graph autoencoders [53], [54], [55], [56] are unsupervised learning frameworks, which aim to encode nodes and/or graphs into the latent space while reconstructing the graph data from the encoded information. The earlier approaches mainly utilize multi-layer perceptrons (MLP) to learn representations. For example, [53] uses the denoising autoencoder to encode and decode the positive pointwise mutual information matrix by MLP. [54] applies the stacked autoencoder to preserve the first-order and second-order similarity of the graph structure. To incorporate the feature information of nodes, [55], [56] use GCN to encode the graph structure information and the node feature information. The decoder rebuilds the adjacency matrix of graph by the nodes' embeddings.

3 Problem Formulation

We begin by introducing the concept of item KG and the KG-based recommendation problem, as well as the notations used in this work.

Interaction Information. In a typical recommendation scenario, we denote the set of users by $\mathcal{U}$, the set of items by $\mathcal{I}$, and all the observed user-item interaction pairs by $\mathcal{O}$. For each user u, we denote the set of items she has interacted by $\mathcal{I}_u^+$ and use $\boldsymbol{e}_u \in \mathbb{R}^{1\times d}$ to denote her embedding, where d denotes the dimensionality of latent space. Moreover, we also define $\mathcal{I}_u^- = \mathcal{I} \setminus \mathcal{I}_u^+$.

Item Knowledge Graph. This work focuses on exploiting the item KG for recommendation instead of building the item KG. Therefore, in this paper, we assume the item KG $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{D}\}$ is available, where $\mathcal{E}$ denotes the set of entities, $\mathcal{R}$ denotes the set of relations, and $\mathcal{D}$ denotes the set of entity-relation-entity triples (h, r, t) describing the KG structure. Here $h \in \mathcal{E}$, $r \in \mathcal{R}$, and $t \in \mathcal{E}$ denote the head entity, relation, and tail entity of a knowledge triple, respectively. For example, the triplet (The Great Gatsby, book.book.author, F.Scott Fitzgerald) states the fact that F.Scott Fitzgerald writes the book "The Great Gatsby". We use $\boldsymbol{e}_h \in \mathbb{R}^{1\times d}$ and $\boldsymbol{e}_r \in \mathbb{R}^{1\times d}$ to denote the embedding of the entity h and relation r. Note that the items are treated as a special type of entities in the KG.

Given the interaction information and item knowledge graph, the recommendation task is to predict whether the user u has potential interests in the item i which she has not interacted with before. More specifically, we aim to learn a

score function $\hat{y}_{ui} = \mathcal{F}(u, i; \mathbf{\Theta}, \mathcal{O}, \mathcal{G})$, where $\hat{y}_{ui}$ denotes the probability that u would like to interact with the item i, and $\mathbf{\Theta}$ denotes the model parameters. Once the score function is learned, we rank candidate items based on the predicted scores in descending order, and choose top-ranked items as recommendations to the user.

4 The Proposed Recommendation Model

Figure 2 shows the structural details of the proposed CGAT model. As shown in Figure 2, CGAT consists of two main components: exploiting KG context module and exploiting interaction graph context module. The probability that a user would like to interact with an item can be predicted based on the user and item representations learned by the above two modules. In the following sections, we will introduce the proposed model in details.

4.1 Exploiting Knowledge Graph Context

To effectively incorporating item KG into recommendation, CGAT exploits KG context of item from two aspects: 1) local context information, and 2) non-local context information.

4.1.1 Local Graph Context

For the entity corresponding to an item, it is always linked with many other entities that can enrich its information in KG. The graph neural networks can be applied to aggregate the feature information of an item's neighbors in the KG. However, directly aggregating neighboring information can not capture user-specific preferences on entities which is important and should be considered on the recommender system. To solve this challenge and identify users' personalized preferences on local graph context, we develop a user-specific graph attention mechanism to aggregate the neighborhood information of an entity in KG. For different users, we compute different attention scores for the same neighborhood entity of item. The embedding of neighborhood entities can then be aggregated based on the user-specific attention scores. Here, we denote the local neighbors of an entity h by $\mathcal{C}_h^l = \{t|(h, r, t) \in \mathcal{D}\}$, and define $\mathcal{C}_h^l$ as the ***local graph context*** of h in KG. Moreover, we also argue that the relations in KG play important roles in understanding semantic information. The neighborhood entities may have different impacts, if they are connected via different relations. To incorporate relation into the attention mechanism, we firstly integrate the embedding of a neighborhood entity $t \in \mathcal{C}_h^l$ and the embedding of corresponding relation r by the following linear transformation,

$$\boldsymbol{e}_{rt} = \big(\boldsymbol{e}_r || \boldsymbol{e}_t\big)\boldsymbol{W}_0, \tag{1}$$

where $||$ is the concatenation operation, $\boldsymbol{W}_0 \in \mathbb{R}^{2d\times d}$ is the weight matrix. The user-specific attention score $\alpha_u(h, r, t)$ that describes the importance of the entity $t \in \mathcal{C}_h^l$ to the entity h, for a target user u, is defined as follows,

$$\alpha_u(h, r, t) = \frac{\exp\big[\pi_u(h, r, t)\big]}{\sum_{(h,\tilde{r},\tilde{t})\in\mathcal{D}} \exp\big[\pi_u(h, \tilde{r}, \tilde{t})\big]}. \tag{2}$$

The operation $\pi_u(h, r, t)$ is performed by a single-layer feed forward neural network, which is defined as follows,

$$\pi_u(h, r, t) = \tanh\big[(\boldsymbol{e}_h || \boldsymbol{e}_{rt})\boldsymbol{W}_1 + \boldsymbol{b}_1\big]\boldsymbol{m}_u^\top, \tag{3}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{2d\times d}$, and $\boldsymbol{b}_1 \in \mathbb{R}^{1\times d}$ are the weight matrix and bias vector respectively, $\boldsymbol{m}_u$ is a non-linear transform of $\boldsymbol{e}_u$ that increases model flexibility, defined as,

$$\boldsymbol{m}_u = \mathrm{ReLU}(\boldsymbol{e}_u\widetilde{\boldsymbol{W}}_1 + \widetilde{\boldsymbol{b}}_1). \tag{4}$$

Here, $\widetilde{\boldsymbol{W}}_1 \in \mathbb{R}^{d\times d}$, and $\widetilde{\boldsymbol{b}}_1 \in \mathbb{R}^{1\times d}$ are the weight matrices and bias vectors respectively. Notice that the attention mechanism will become user-specific by the usage of $\boldsymbol{e}_u$ when computing attention scores, which can reflect the user personalized preference on neighbors. Given the normalized coefficient of each neighboring entity of h, we compute the linear combination of their embedding to obtain the local neighborhood embedding of h as follows,

$$\boldsymbol{e}_{\mathcal{C}_h^l} = \sum_{t\in\mathcal{C}_h^l} \alpha_u(h, r, t)\boldsymbol{e}_t. \tag{5}$$

Then, we aggregate the embedding of entity h and it's local neighborhood embedding $\boldsymbol{e}_{\mathcal{C}_h^l}$ to form a local contextual embedding $\boldsymbol{c}_h^l$ for h as follows,

$$\boldsymbol{c}_h^l = \tanh\big[(\boldsymbol{e}_h || \boldsymbol{e}_{\mathcal{C}_h^l})\boldsymbol{W}_2 + \boldsymbol{b}_2\big], \tag{6}$$

where $\boldsymbol{W}_2 \in \mathbb{R}^{2d\times d}$ and $\boldsymbol{b}_2 \in \mathbb{R}^{1\times d}$ are the weight matrix and bias vector of the aggregator.

4.1.2 Non-Local Graph Context

The user-specific graph attention network explicitly aggregates the local neighbor (one-hop) information of a target entity to enrich the representation of the target entity. Exiting methods aggregate the non-local context by feature propagation layer by layer [15], [16]. However, this can not directly capture feature information of non-local context and the effects of high-order neighbors will be weakened, even some "noise" entities may be brought into context. To offset this gap, we propose a biased random walk based Gated Recurrent Unit (GRU) module to aggregate non-local context information of entities in KG.

Firstly, we design a biased random walk sampling (BRWS) strategy to explicitly extract the non-local context of a target entity h. To achieve a wider depth-first search, we repeat biased random walk from h to obtain M paths, which have a fixed length L. The walk iteratively travels to the neighbors of current entity with a probability p, which is defined as follows,

$$p(t_{k+1}) = \begin{cases} \gamma & \text{if } t_{k+1} \in \mathcal{C}_{t_{k-1}}^l \text{or } t_{k+1} = t_{k-1}, \\ 1-\gamma & \text{else,} \end{cases} \tag{7}$$

where t_k is the k-th entity of a path, t_0 denotes the root entity h. To encourage wider search, we empirically set $0 < \gamma < 0.5$. After obtaining the M paths and $M * L$ entities by walk, we sort entities according to their frequency in walks in descending order, and choose a set of top-ranked entities orderly. These entities are defined as the ***non-local graph context*** of the entity h in KG, and denoted by $\mathcal{C}_h^g$. Note that $\mathcal{C}_h^g$ may also include some first-order neighbors of h (refer to Section 5.5 for more discussions). In the experiments, we set $|\mathcal{C}_h^g| = |\mathcal{C}_h^l|$ to reduce the number of hyper-parameters of CGAT, and empirically set the parameters γ, M, and L to 0.2, 15, and 8 respectively, based on the model performances on the validation data.

After sampling, we employ GRU to model the dependency between an entity h and its non-local context $\mathcal{C}_h^g$. GRU is a variant of RNN model, which has been successfully used for various sequential tasks. We argue that $\mathcal{C}_h^g$ can be seen as a frequency sequence data. Indeed, the more frequently an entity appears in random walks, the more important it is to the target entity h. Based on this intuition, we input $\mathcal{C}_h^g$ into GRU in reverse order, which is formulated as follows,

$$\begin{aligned}
\boldsymbol{z}_t &= \sigma((\boldsymbol{h}_{t-1}||\boldsymbol{e}_t)\boldsymbol{W}_3^z), \\
\boldsymbol{r}_t &= \sigma((\boldsymbol{h}_{t-1}||\boldsymbol{e}_t)\boldsymbol{W}_3^r), \\
\widetilde{\boldsymbol{h}}_t &= \tanh(((\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})||\boldsymbol{e}_t)\boldsymbol{W}_3^o), \\
\boldsymbol{h}_t &= (1-\boldsymbol{z}_t)\odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \widetilde{\boldsymbol{h}}_t,
\end{aligned} \tag{8}$$

where $\boldsymbol{h}_t$ is the output hidden state of t-th entity, $\odot$ denotes the Hadamard product, $\sigma(\cdot)$ denotes the sigmoid function, $\boldsymbol{W}_3^j (j \in \{z, r, o\})$ are weight matrices. We use the last step output of GRU as the embedding of $\mathcal{C}_h^g$, which is denoted as follows,

$$\boldsymbol{e}_{\mathcal{C}_h^g} = \text{GRU}(\overleftarrow{\mathcal{C}_h^g}), \tag{9}$$

where $\overleftarrow{\mathcal{C}_h^g}$ denotes the reverse set of $\mathcal{C}_h^g$. Then, we aggregate $\boldsymbol{e}_h$ and $\boldsymbol{e}_{\mathcal{C}_h^g}$ to form the non-local contextual embedding $\boldsymbol{c}_h^g$ for h as follows,

$$\boldsymbol{c}_h^g = \tanh\big[(\boldsymbol{e}_h||\boldsymbol{e}_{\mathcal{C}_h^g})\boldsymbol{W}_2 + \boldsymbol{b}_2\big]. \tag{10}$$

Here, we use the same aggregator parameters as in Eq (6).

The proposed non-local graph context extraction strategy has the following two advantages. Firstly, the non-local neighbors of a target entity h are selected based on their occurrence frequency in the sampling paths. Thus, it can filter out some "noise" entities with few occurrences, which may not be relevant to the target entity h. Secondly, the important non-local neighbors are sorted based on the sampled frequency. GRU is then used to explicitly aggregate the sequence of important non-local neighbors (*i.e.*, the non-local graph context $\mathcal{C}_h^g$) of the target entity h, to build direct interactions between the target entity and its non-local graph context, instead of using the layer-to-layer feature propagation. Note that we input $\mathcal{C}_h^g$ into GRU in reverse order. Thus, in $\mathcal{C}_h^g$, the nodes with larger sampled frequency tend to have larger influences on the target node h. In this way, the influences of high-order but important neighbors will not be weakened.

Given the embeddings of the local and non-local context of h in KG, we apply a gate mechanism to adaptively integrate these two embeddings by learning the weights in each dimension as,

$$\boldsymbol{c}_h = \sigma(\boldsymbol{\omega}) \odot \boldsymbol{c}_h^l + (1-\sigma(\boldsymbol{\omega})) \odot \boldsymbol{c}_h^g, \tag{11}$$

where $\boldsymbol{\omega} \in \mathbb{R}^{1\times d}$ is a learnable vector, $\sigma(\cdot)$ denotes the sigmoid function. As items are a special type of entities in KG, we can use Eq. (11) to compute the context embedding $\boldsymbol{c}_i$ of item i, considering its local context $\mathcal{C}_i^l$ and non-local context $\mathcal{C}_i^g$ in KG. Then, considering the original item information, we concatenate $\boldsymbol{e}_i$ and $\boldsymbol{c}_i$ to obtain the contextualized representation of an item i as follows,

$$\boldsymbol{q}_i = (\boldsymbol{e}_i||\boldsymbol{c}_i). \tag{12}$$

Algorithm 1 CGAT Optimization Algorithm

Input: Observed interactions $\mathcal{O}$, item knowledge graph $\mathcal{G}$
Output: Score function $\mathcal{F}(u, i; \boldsymbol{\Theta}) = \hat{y}_{ui}$

```
Randomly initialize all parameters
Construct the set Õ and D̃ based on O and D;
for iter = 1, 2, ··· , max_iter do
    Sample a batch of tuples B_1 from Õ;
    Sample a batch of tuples B_2 from D̃;
    Compute gradients of Eq. (21) with respect to Θ by back-propagation, based on tuples in B_1 and B_2;
    Update Θ by gradient descent algorithm (i.e., Adam) with learning rate η;
end for
return F(u, i; Θ)
```

4.2 Exploiting Interaction Graph Context

In practice, a user's historical items are usually used to describe her potential interests [1]. For example, the classical SVD++ model [23] treats a user u's historical items $\mathcal{I}_u^+$ as the implicit feedback given by u, and models the influences of $\mathcal{I}_u^+$ on a target item i for recommendation. However, most KG based methods ignore the influence of target item when utilizing the historical items to represent user preference. To solve the third challenge, we firstly define $\mathcal{I}_u^+$ as the ***interaction graph context*** of user u, and then develop an item-specific attention mechanism to model the influences of i on $\mathcal{I}_u^+$. The basic assumption is that a user's historical item may have different importance in estimating her preferences on different candidate items. For each item $j \in \mathcal{I}_u^+$, its relevance weight with respect to the target item i is defined as follows,

$$\beta(i,j) = \frac{\exp\big[\tanh\big((\boldsymbol{q}_i||\boldsymbol{q}_j)\boldsymbol{w}^\top + b\big)\big]}{\sum_{k\in\mathcal{I}_u^+}\exp\big[\tanh\big((\boldsymbol{q}_i||\boldsymbol{q}_k)\boldsymbol{w}^\top + b\big)\big]}, \tag{13}$$

where $\boldsymbol{w} \in \mathbb{R}^{1\times 4d}$ is a weight vector, b is the bias, $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ are the contextualized representations of items i and j. Then, we define the embedding of the graph context $\mathcal{I}_u^+$, with respect to a target item i, as follows,

$$\boldsymbol{e}_{\mathcal{I}_u^+}^i = \sum_{j\in\mathcal{I}_u^+}\beta(i,j)\boldsymbol{q}_j. \tag{14}$$

A non-linear transformation is then used to aggregate $\boldsymbol{e}_u$ and $\boldsymbol{e}_{\mathcal{I}_u^+}$ to form the contextual embedding for u as follows,

$$\boldsymbol{c}_u = \text{ReLU}\big[(\boldsymbol{e}_u||\boldsymbol{e}_{\mathcal{I}_u^+}^i)\boldsymbol{W}_3 + \boldsymbol{b}_3\big], \tag{15}$$

where $\text{ReLU}(\cdot)$ is the activation function, $\boldsymbol{W}_3 \in \mathbb{R}^{3d\times d}$ and $\boldsymbol{b}_3 \in \mathbb{R}^{1\times d}$ are the weight matrix and the bias vector. Considering the original user information, we concatenate $\boldsymbol{e}_u$ and $\boldsymbol{c}_u$ to form the contextualized representation for u as follows,

$$\boldsymbol{p}_u = (\boldsymbol{e}_u||\boldsymbol{c}_u). \tag{16}$$

Finally, given the contextualized representations of user u and item i, the prediction of u's preference on i can be defined as follows,

$$\hat{y}_{ui} = \boldsymbol{p}_u\boldsymbol{q}_i^\top. \tag{17}$$

4.3 Model Training

The Bayesian personalized ranking (BPR) optimization criterion [27] is used to learn the model parameters of CGAT. BPR assumes that the interacted items should have higher ranking scores than the un-interacted items for each user. Here, we define the BPR loss as follows,

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i^+,i^-)\in\widetilde{\mathcal{O}}} -\log\sigma(\hat{y}_{ui^+} - \hat{y}_{ui^-}), \tag{18}$$

where $\widetilde{\mathcal{O}}$ is constructed by negative sampling. Empirically, for each $(u,i)\in\mathcal{O}$, we randomly sample 5 items from $\mathcal{I}\backslash\mathcal{I}_u^+$ in the experiments. As we also need to learn the embedding of entities and relations in KG, we design a regularization loss based on the KG structure in order to avoid over-fitting and capture graph topology information. Specifically, for each triple $(h,r,t)\in\mathcal{D}$, we first define the following score to describe the distance between the head entity h and the tail entity t via relation r in the latent space,

$$s_r(h,t) = ||\boldsymbol{e}_h - \boldsymbol{e}_{rt}||_2^2, \tag{19}$$

where $\boldsymbol{e}_{rt}$ is obtained by Eq. (1). Then, we define the regularization loss as follows,

$$\mathcal{L}_{\text{KG}} = \sum_{(h,r,t,t')\in\widetilde{\mathcal{D}}} \log\sigma\big(s_r(h,t) - s_r(h,t')\big), \tag{20}$$

where $\widetilde{\mathcal{D}}$ is constructed by randomly sampling an entity t' from $\mathcal{E}\setminus\mathcal{C}_h^l$, for each $(h,r,t)\in\mathcal{D}$. The motivation is that, in the latent space, the distance between an entity h and its directly connected neighbor t should be smaller than the distance between h and the entity t' that is not directly connected to h, via relation r. Then, the model parameters can be learned by solving the following objective function,

$$\min_{\boldsymbol{\Theta}} \mathcal{L}_{\text{BPR}} + \lambda_1\mathcal{L}_{\text{KG}} + \lambda_2||\boldsymbol{\Theta}||_2^2, \tag{21}$$

where $\boldsymbol{\Theta}$ denotes all the parameters of CGAT, λ_1 and λ_2 are the regularization parameters. The problem in Eq. (21) is solved by a gradient descent algorithm. The details of the optimization algorithm are summarized in Algorithm 1.

In our implementation, we randomly sample S neighbors from $\mathcal{C}_h^l$ for a target entity h, and N historical items from $\mathcal{I}_u^+$ for a target user u, to compute the attention weights defined in Eq. (2) and Eq. (13) respectively. This trick can help keep the computational pattern of each mini-batch fixed and improve computation efficiency. Moreover, we also set the size of non-local context $|\mathcal{C}_h^g|$ to S. In model training, S and N are fixed. Let B denote the number of sampled user-item interactions in each batch. The time complexity of the BRWS procedure is $O(|\mathcal{I}|SML)$, which can be performed before training. In each iteration, to exploit KG context, the user-specific graph attention mechanism and the GRU module have computational complexity $O(BNSd^2)$. The complexity of exploiting the interaction graph context is $O(BNd^2)$. The overall complexity of each minibacth iteration is $O\big(B(NSd^2+Nd^2)\big)\approx O(BNSd^2)$, which is linear with all hyper-parameters except for d.

TABLE 1
Statistics of the experimental datasets.

	FM	ML	BC	DF
#Users	1,872	6,036	17,860	63,566
#Items	3,846	2,347	14,967	1,362
#Interactions	21,173	376,886	69,876	3,257,131
#Density	0.29%	2.66%	0.026%	3.76%
#Entities	9,366	7,008	77,903	28,115
#Relations	60	7	25	7
#Triples	15,518	20,782	151,500	160,519

5 Experiments

In this section, we conduct empirical experiments on real datasets to answer the following research questions:

- **(RQ1)** Whether does CGAT achieve better performances than state-of-the-art KG-based recommendation methods on different datasets?
- **(RQ2)** How do different components of CGAT, *e.g.*, exploiting local context or user-specific attention mechanism, affect the model performance?
- **(RQ3)** How does CGAT perform over different user groups with different interaction sparsity level, compared with other GNN-based baseline methods?
- **(RQ4)** Whether the BRWS module can capture the non-local graph context?
- **(RQ5)** How do various hyper-parameters, *e.g.*, the size of sampled neighbors and the dimensionality of latent space, would like to impact the model performances?

5.1 Experimental Settings

5.1.1 Datasets

The experiments are performed on the following datasets: Last-FM[3], Movielens-1M[4], Book-Crossing[5], and Dianping Food[6] (respectively denoted by FM, ML, BC, and DF). The first three are public datasets, and the last one is from Meituan-Dianping Group. Following [12], [15], [25], we keep all the ratings on FM and BC datasets as observed implicit feedback, due to data sparsity. For ML dataset, we keep ratings larger than 4 as implicit feedback. On DF dataset, we only keep users that have at least 30 interaction items for experiments, due to the limitation of computation resources. The KGs of the FM, ML, and BC datasets are constructed by Microsoft Satori. As introduced in [25], only the triples from the whole KG with a confidence level greater than 0.9 are retained. The sizes of ML and BC KGs are further reduced by only selecting the triples where the relation name contains "film" and "book", respectively. For these three datasets, we match the items and entities in sub-KGs by their names (*e.g.*, head, film.film.name, tail for ML). The items matching no entities or multiple entities are removed. The KG for the DF dataset is built by the internal toolkit of Meituan-Dianping Group. In this KG, there are 8 types of entities, including Point-of-Interest (*i.e.*, restaurant), city, first-level and second-level category, star, business area, dish, and tag. More details about the DF dataset can be found in [15]. The statistics of these experimental datasets are summarized in Table 1. The interaction data of DF

3. https://grouplens.org/datasets/hetrec-2011/
4. https://grouplens.org/datasets/movielens/1m/
5. http://www2.informatik.uni-freiburg.de/~cziegler/BX/
6. https://www.dianping.com/

TABLE 2
Performances of different recommendation algorithms. The best results are in **bold faces** and the second best results are underlined. * indicates CGAT significantly outperforms the competitors with $p < 0.05$ using Wilcoxon signed rank significance test.

Datasets	Methods	P@10	R@10	HR@10	P@20	R@20	HR@20	P@50	R@50	HR@50
FM	GraphSAGE	0.0268	0.1150	0.2211	0.0214	0.1831	0.3319	0.0142	0.2975	0.4888
	FMG	0.0251	0.1026	0.2096	0.0216	0.1814	0.3313	0.0138	0.2912	0.4615
	MCRec	0.0412	0.1646	0.3247	0.0320	0.2562	0.4627	0.0203	0.4099	0.6402
	CFKG	0.0280	0.1168	0.2362	0.0222	0.1857	0.3404	0.0135	0.2812	0.4773
	RippleNet	0.0285	0.1214	0.2423	0.0229	0.1948	0.3628	0.0157	0.3260	0.5336
	MKR	0.0278	0.1162	0.2356	0.0215	0.1820	0.3356	0.0138	0.2877	0.4809
	KGNN-LS	0.0284	0.1186	0.2441	0.0216	0.1824	0.3398	0.0136	0.2828	0.4809
	KGAT	0.0460	0.1879	0.3634	0.0351	0.2881	0.5027	0.0206	0.4158	0.6432
	CGAT	**0.0512***	**0.2106***	**0.4022***	**0.0369***	**0.2994***	**0.5203***	**0.0218***	**0.4413***	**0.6687***
ML	GraphSAGE	0.0749	0.0753	0.4814	0.0630	0.1211	0.6246	0.0510	0.2346	0.7858
	FMG	0.1081	0.1039	0.5580	0.0920	0.1714	0.6968	0.0649	0.2865	0.8150
	MCRec	0.1125	0.1199	0.6035	0.0927	0.1884	0.7316	0.0668	0.3195	0.8486
	CFKG	0.1054	0.1038	0.5680	0.0896	0.1753	0.7126	0.0633	0.2991	0.8388
	RippleNet	0.1271	0.1251	0.6227	0.1043	0.2008	0.7474	0.0758	0.3442	0.8667
	MKR	0.1376	0.1370	0.6581	0.1154	0.2192	0.7765	0.0848	0.3793	0.8852
	KGNN-LS	0.1311	0.1310	0.6419	0.1126	0.2172	0.7766	0.0833	0.3762	0.8811
	KGAT	0.1533	0.1608	0.7090	0.1274	0.2541	0.8179	0.0910	0.4189	0.9066
	CGAT	**0.1575***	**0.1674***	**0.7219***	**0.1288***	**0.2608***	**0.8264***	**0.0916***	**0.4311***	**0.9191***
BC	GraphSAGE	0.0092	0.0410	0.0854	0.0064	0.0554	0.1138	0.0044	0.0876	0.1781
	FMG	0.0149	0.0696	0.1349	0.0101	0.0909	0.1781	0.0059	0.1254	0.2368
	MCRec	**0.0161**	**0.0764**	**0.1450**	0.0106	**0.0947**	0.1803	0.0064	0.1328	0.2518
	CFKG	0.0155	0.0725	0.1391	0.0101	0.0904	0.1745	0.0061	0.1291	0.2435
	RippleNet	0.0147	0.0706	0.1336	0.0099	0.0880	0.1736	0.0060	0.1261	0.2429
	MKR	0.0154	0.0732	0.1386	0.0105	0.0920	0.1811	0.0063	0.1306	0.2496
	KGNN-LS	0.0155	0.0730	0.1411	0.0104	0.0910	0.1797	0.0062	0.1306	0.2454
	KGAT	0.0146	0.0615	0.1308	0.0105	0.0830	0.1739	0.0068	0.1274	0.2518
	CGAT	**0.0161**	0.0645	0.1402	**0.0119***	0.0920	**0.1909***	**0.0078***	**0.1412***	**0.2718***
DF	GraphSAGE	0.0877	0.0912	0.5710	0.0754	0.1554	0.7509	0.0588	0.2986	0.9194
	FMG	0.0559	0.0575	0.4164	0.0494	0.1008	0.6021	0.0442	0.2235	0.8515
	MCRec	0.0841	0.0866	0.5509	0.0726	0.1478	0.7288	0.0570	0.2877	0.9059
	CFKG	0.0832	0.0857	0.5484	0.0719	0.1466	0.7285	0.0566	0.2860	0.9064
	RippleNet	0.0955	0.099	0.6004	0.0818	0.1683	0.7756	0.0626	0.3182	0.9294
	MKR	0.1078	0.1115	0.6454	0.0912	0.1869	0.8102	0.0688	0.3479	0.9466
	KGNN-LS	0.0816	0.0836	0.5402	0.0706	0.1438	0.7204	0.0559	0.2825	0.9032
	KGAT	0.1101	0.1142	0.6548	0.0936	0.1926	0.8191	0.0706	0.3573	0.9503
	CGAT	**0.1200***	**0.1242***	**0.6867***	**0.1001***	**0.2053***	**0.8415***	**0.0737***	**0.3725***	**0.9569***

dataset and the KGs of the four datasets are currently public available at https://github.com/hwwang55.

5.1.2 Setup and Metrics

For each dataset, we randomly select 60% of the observed user-item interactions for model training, and choose another 20% of interactions for parameter tuning. The remaining 20% of interactions are used as testing data. This setting has been widely used in previous research works [12], [15], [25]. The quality of the top-K item recommendation is assessed by three widely used evaluation metrics: Precision@K, Recall@K, and Hit Ratio@K (respectively denoted by P@K, R@K, and HR@K). In the experiments, we empirically set K to 10, 20, and 50. For each metric, we first compute the accuracy for each testing user, and then report the averaged accuracy over all testing users.

5.1.3 Baseline Methods

To demonstrate the effectiveness, we compare CGAT with the following baseline methods:

- **GraphSage** [48]: This is a representative graph representation learning method. In this method, we connect the user-item interaction graph with the item KG to form a large heterogeneous graph. The GRU aggregator is applied to aggregate the neighborhood information of the user and item. Moreover, the BPR loss is used to model the user-item interactions for recommendation.
- **FMG** [9]: This is a HIN-based rating prediction method. We replace its optimization objective function by BPR loss function for top-N item recommendation.
- **MCRec** [10]: This HIN-based method employs the co-attention mechanism to leverage the context information extracted from meta-path for item recommendation.
- **CFKG** [11]: This method integrates the multi-type user behaviors and item KG into a unified graph, and employs TransE [57] to learn the entity embedding.
- **RippleNet** [12]: This method exploits KG information by propagating a user's preferences over the set of entities along paths in KG rooted at her historical items.
- **MKR** [25]: This is a multi-task feature learning approach that uses KG embedding task to assist the recommendation task, where a cross&compress unit is designed to approximate the high-order feature interactions.
- **KGNN-LS** [15]: This approach applies GCN on item KG to compute the item embedding by propagating and

TABLE 3
The recommendation performances achieved by KGAT and CGAT variants on different datasets.

Datasets	Metrics	KGAT	$\text{CGAT}_{/\text{L}}$	$\text{CGAT}_{/\text{G}}$	$\text{CGAT}_{/\text{UA}}$	CGAT_{RWR}	CGAT_{GRU}	CGAT_{ATT}	CGAT
FM	P@20	0.0351	0.0363	0.0370	0.0366	**0.0372**	0.0362	**0.0372**	0.0369
	R@20	0.2881	0.2949	0.2981	0.2964	0.2999	0.2955	**0.3023**	0.2994
	HR@20	0.5027	0.5118	0.5167	0.5136	0.5118	0.5064	**0.5203**	**0.5203**
ML	P@20	0.1274	**0.1292**	0.1230	0.1291	0.1222	0.1231	0.1248	0.1288
	R@20	0.2541	0.2559	0.2432	0.2563	0.2404	0.2388	0.2477	**0.2608**
	HR@20	0.8179	0.8193	0.8111	0.8215	0.8070	0.8035	0.8152	**0.8264**
BC	P@20	0.0105	0.0116	0.0116	0.0116	0.0118	0.0115	0.0093	**0.0119**
	R@20	0.0830	**0.0924**	0.0897	0.0879	0.0893	0.0880	0.0736	0.0920
	HR@20	0.1739	0.1884	0.1864	0.1817	0.1881	0.1836	0.1564	**0.1909**

aggregating the neighborhood information on item KG. The user's personalized preferences on relations and label smoothness regularization are also considered.

- **KGAT** [16]: This approach employs graph attention mechanism on a unified graph integrating item KG and interaction graph to exploit the graph context for recommendation.

5.1.4 Implementation Details

For CGAT, the dimensionality of latent space d is chosen from $\{4, 8, 16, 32, 64, 128\}$. The number of local neighbors of an entity S and the number of a user's historical items N used in model training are selected from $\{2, 4, 8, 16, 24, 32, 40\}$. The regularization parameters λ_1 and λ_2 are chosen from $\{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 10^{-2}\}$. We implement CGAT by Pytorch, and the Adam optimizer [58] is used to learn the model parameters. The learning rate η is chosen from $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$. The hyper-parameters of baseline methods are set following original papers. For all methods, the optimal hyper-parameters are determined based on the performances on the validation data.

5.2 Performance Comparison (RQ1)

Table 2 summarizes the comparison results on different datasets. We make the following observations.

- On FM, ML, and DF datasets, KGAT usually achieves the best performances among all baselines. This verifies that aggregating context information by graph attention mechanism in KG is of importance to enrich item representation. However, the performance of KGNN-LS is not significant. The reason may be that only considering relation to aggregate entities' context can not differentiate the importance of different neighbors. Moreover, the better performance of CGAT than these GNN-based methods also verifies the significance of solving the three challenges we raised.
- On BC dataset, the path-based method MCRec usually outperforms other baseline methods, and achieves better top-10 recommendation accuracy than CGAT. The KG and interaction graphs on BC dataset are very sparse. MCRec employs a priority based sampling technique to select the high-quality meta-paths in HIN and employs co-attention mechanism to enhance the representations of users, items, and the meta-path context. Thus, MCRec can effectively utilize the heterogeneous information to solve the data sparsity problem.
- Moreover, GraphSage performs the worst among baselines on ML and BC datasets, and achieves the second worst performances on FM dataset. The potential reasons are as follows. Firstly, in GraphSage, the relation information in KG is not considered in learning the node representations. Secondly, the GRU aggregator of GraphSage operates on a random permutation of the first-hop neighbors of a target node in KG, thus it cannot differ the importance of different neighboring nodes. Differing from GraphSage, KGAT and CGAT use relation-aware attention mechanism to differ the importance of neighboring nodes, thus they can usually achieve better performances than GraphSage.
- On all datasets, CGAT usually achieves the best performances, in terms of all metrics. In most of the scenarios (*i.e.*, 32 among 36 evaluation metrics), the proposed CGAT method significantly outperforms baseline methods with $p < 0.05$, using the Wilcoxon signed rank significance test [59]. Over all datasets, on average, CGAT outperforms GraphSage, FMG, MCRec, CFKG, RippleNet, MKR, KGNN-LS, and KGAT by 42.22%, 30.65%, 11.69%, 23.43%, 18.11%, 17.68%, 20.64%, 4.26%, respectively, in terms of HR@20. These results demonstrate the effectiveness of CGAT in exploiting both the KG context and users' historical interaction context for recommendation.

5.3 Ablation Study (RQ2)

To investigate the importance of each component of CGAT, we conduct ablation studies to evaluate the performances of the following CGAT variants:

- $\textbf{CGAT}_{\textbf{/L}}$ deletes the local context embedding of item from CGAT and only considers the non-local context embedding as final context embedding, *i.e.*, the coefficient $\sigma(\boldsymbol{\omega})$ in Eq.(11) is set to $\mathbf{0}$;
- $\textbf{CGAT}_{\textbf{/G}}$ removes the non-local context embedding of item from original model, which is contrast to $\textbf{CGAT}_{/\text{L}}$ model, *i.e.*, the coefficient $\sigma(\boldsymbol{\omega})$ in Eq.(11) is set to $\mathbf{1}$;
- $\textbf{CGAT}_{\textbf{/UA}}$ removes the user's embedding in exploiting the local context information in KG (*i.e.*, removing $\boldsymbol{m}_u$ in Eq. (3)) for evaluating the effectiveness of user-specific attention mechanism.
- $\textbf{CGAT}_{\textbf{RWR}}$: In this variant, we replace the BRWS sampling method by the Random Walk with Restart (RWR) algorithm [60]. The restart probability c is set to 0.5, which is the optimal setting. As the default settings for the number of paths M and the length of the path L are 15 and 8 respectively, BRWS samples 120 nodes from the neighbourhood of a target entity. For fair comparison, we

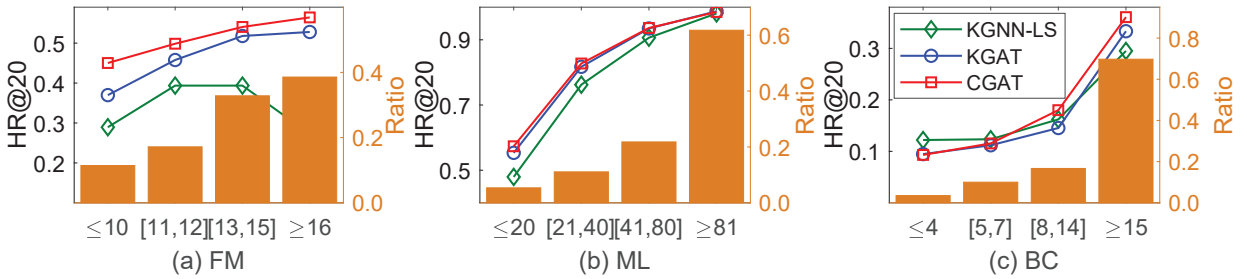


Fig. 3. Performance comparison at different interaction sparsity levels of user groups on different datasets. The histogram indicates the ratio of total number of testing users. The lines show the performances of different methods in terms of HR@20.

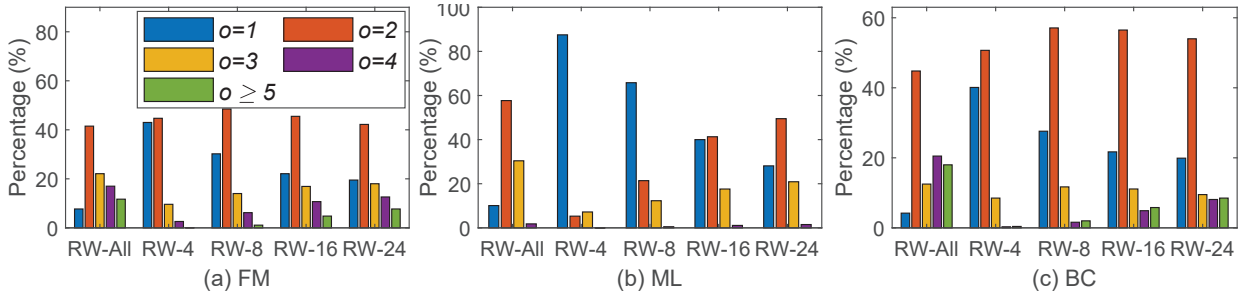


Fig. 4. The order distribution of the sampled nodes in BRWS. The diameters of the KGs of FM, ML, and BC datasets are 16, 4, and 16 respectively.

start the RWR from a target entity h and sample 120 nodes from its neighbourhood. Then, we sort the sampled nodes in descending order according to the sampled frequency and choose the same number of top-ranked nodes as used in BRWS to build the non-local context of h.

- **CGAT$_{\text{GRU}}$**: In this variant, we use GRU to aggregate the local graph context of an entity in KG. Following [48], we adapt GRU to operate on an unordered set by simply applying the GRU to a random permutation of the target node's local-context neighbours.
- **CGAT$_{\text{ATT}}$**: In this variant, we use the attention mechanism defined in Eq. (2) to aggregate the non-local context information and replace $\boldsymbol{e}_{rt}$ by $\boldsymbol{e}_t$ to define the attention score in Eq. (3).

The performances of CGAT variants on different datasets are summarized in Table 3. We have the following findings.

- CGAT consistently outperforms the variants CGAT$_{/\text{L}}$ and CGAT$_{/\text{G}}$ in terms of most evaluation metrics, indicating both local and non-local context in KG are essential for recommendation. The local context enrich item representation by first-order neighbors information while capturing user preferences for entities. The non-local context selected by biased random walk provides item fruitful and filtered high-order neighbors information.
- CGAT$_{/\text{L}}$ is slightly superior than CGAT$_{/\text{G}}$ on ML and BC datasets. This indicates that non-local context information plays a complementary role to the local context information, and sometimes may be more important than local context information in improving the recommendation accuracy. Moreover, we can also note that CGAT$_{/\text{L}}$ consistently outperforms KGAT on all datasets in terms of all three metrics. This observation again demonstrates the effectiveness of the proposed random walk based non-local context extraction strategy for recommendation.
- CGAT achieves better performance than CGAT$_{/\text{UA}}$. This demonstrates the user-specific graph attention mechanism is more suitable for personalized recommendation than simple attention mechanism that can not capture users' personalized preferences.
- CGAT$_{\text{RWR}}$ and CGAT achieves better performances than KGAT on FM and BC datasets, and CGAT achieves the best results on ML and BC datasets. This demonstrates the effectiveness of the proposed recommendation framework. Moreover, CGAT outperforms CGAT$_{\text{RWR}}$ on ML and BC datasets, and achieves comparable results with CGAT$_{\text{RWR}}$ on FM dataset. This indicates the proposed BRWS strategy is more effective in capturing the non-local context for recommendation.
- In addition, CGAT outperforms CGAT$_{\text{GRU}}$ on all datasets, in terms of all metrics. This indicates the relation aware attention mechanism is more suitable to aggregate the local information. Compared with CGAT$_{\text{ATT}}$, CGAT achieves better performances on ML and BC datasets and achieves comparable results on FM dataset. This indicates the GRU mechanism is more suitable to aggregate the non-local context information that includes sequential information between nodes.

TABLE 4
Performances of CGAT with respect to different settings of M measured by HR@20.

Datasets	$M=5$	$M=10$	$M=15$	$M=20$	$M=25$
FM	0.5167	0.5130	**0.5203**	0.5112	0.5161
ML	0.8254	0.8250	**0.8264**	0.7990	0.8232
BC	0.1853	0.1814	**0.1909**	0.1797	0.1811

TABLE 5
Performances of CGAT with respect to different settings of L measured by HR@20.

Datasets	$L=4$	$L=8$	$L=12$	$L=16$	$L=20$
FM	0.5161	**0.5203**	0.5191	0.5106	0.5191
ML	0.8222	**0.8264**	0.8235	0.8233	0.8259
BC	**0.1923**	0.1909	0.1828	0.1825	0.1861

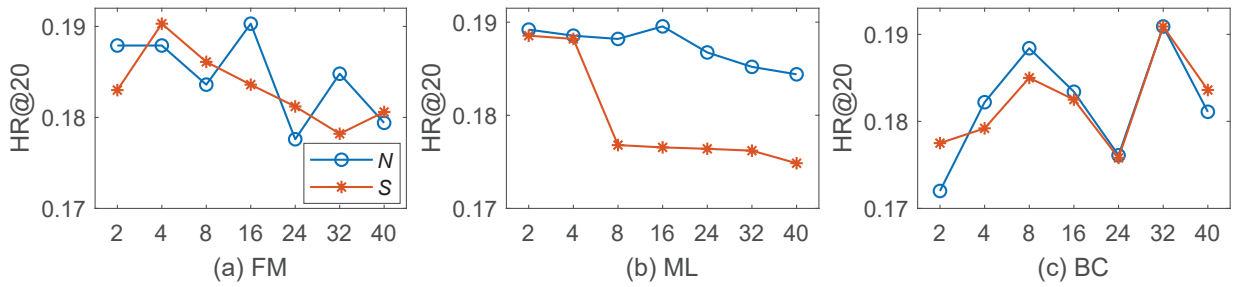


Fig. 5. Performances of CGAT on different datasets with respect to different settings of N and S measured by HR@20.

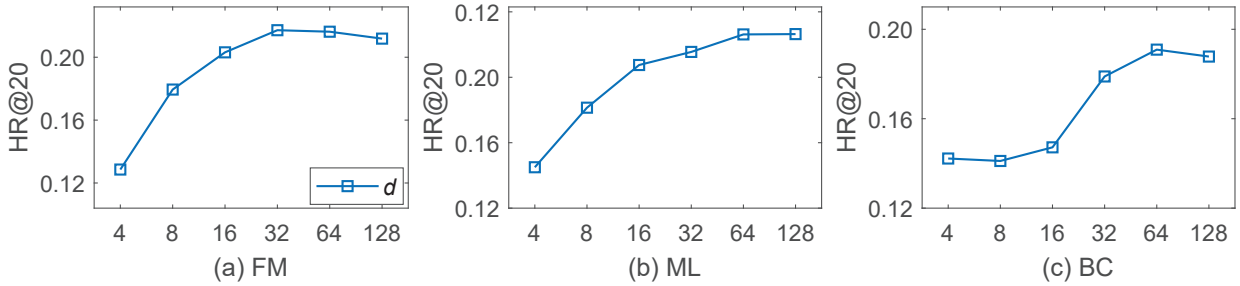


Fig. 6. Performances of CGAT on different datasets with respect to different settings of d measured by HR@20.

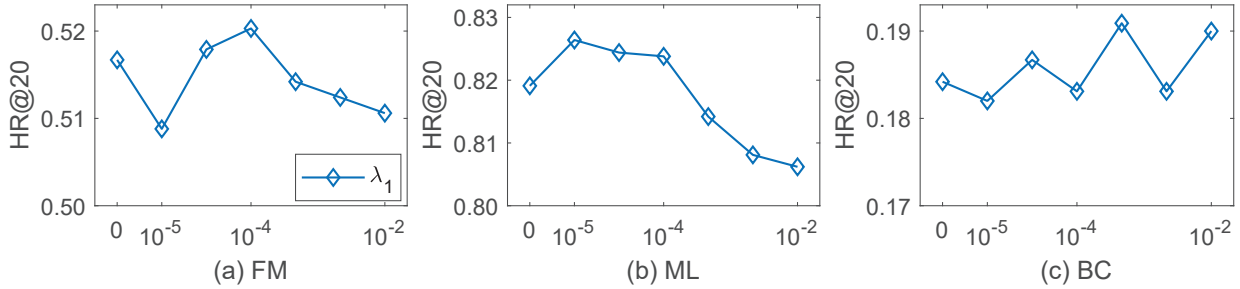


Fig. 7. Performances of CGAT on different datasets with respect to different settings of λ_1 measured by HR@20.

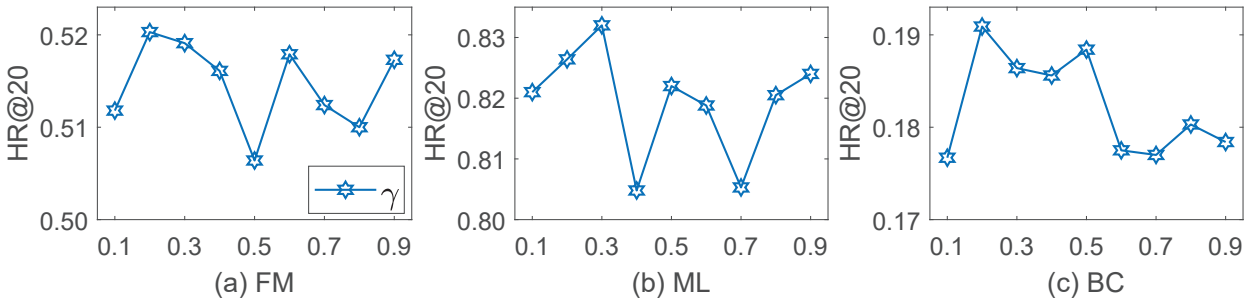


Fig. 8. Performances of CGAT on different datasets with respect to different settings of γ measured by HR@20.

5.4 Interaction Sparsity Study (RQ3)

One advantage of incorporating knowledge graph into recommendation is that it may help alleviate the sparsity issue, which usually limits the recommendation performances. In the scenarios with few interactions, it is hard to learn the optimal representations for prediction. Therefore, we also investigate the ability of CGAT for alleviating the data sparsity problem.

Towards this end, we compare the performances of CGAT and other two GNN-based methods (*i.e.*, KGNN-LS and KGAT) over different user groups with different sparsity levels. Particularly, we divide the testing users into four non-overlapping groups based on the number of interaction items per user. Meanwhile, we also try to keep each group has a reasonable number of users. Take ML dataset as an example, the grouped boundaries of interaction item numbers are 20, 40, and 80, which means that the users with at most 20 interaction items are in the first group, and users with the number of interaction items between 21 and 40 are in the second group, and so on. Figure 3 shows the recommendation performances measured by HR@20 on different user groups. We can note that CGAT usually outperforms KGNN-LS and KGAT on all groups in different datasets, especially on sparser user groups in FM and ML datasets. This demonstrates the effectiveness of CGAT in the sparse interaction scenarios. The potential reason is that CGAT aggregates local graph context by considering the

user's personalized preferences and aggregates non-local graph context by biased random walk sampling, which alleviates the problems of incompletion and noise of the item knowledge graph.

5.5 Effectiveness of BRWS (RQ4)

The proposed method utilizes biased random walk sampling to capture the high-order neighborhood information in item KG. Here, we define the order (denoted by o) of a sampled entity as its distance to the staring entity of the random walk in the KG. Figure 4 shows the order distributions of the sampled nodes on different datasets, by empirically setting M and L to 15 and 8 respectively. In Figure 4, RW-All denotes the order distribution for all sampled entities, and RW-4, RW-8, RW-16, and RW-24 denote the distributions for the top-ranked 4, 8, 16, and 24 entities respectively. In total, 120 entities can be sampled. As shown in Figure 4, 50.8%, 32.2%, ad 51.0% of the sampled unique entities in FM, ML, and BC datasets are with order larger than 2 (*i.e.*, $o > 2$). Moreover, in CGAT, only the top-ranked sampled entities can be used as the non-local context information in KGs. From Figure 4, we can note that 69.8%, 34.2%, and 72.4% of the top-8 sampled entities (*i.e.*, RW-8) used by CGAT to exploit the non-local graph context are with order $o \geq 2$. By using more sampled entities as non-local graph context, more high-order neighbors would be directly exploited in CGAT. These results demonstrate that CGAT can effectively exploit the high-order neighborhood entities in KGs for recommendation. Moreover, Table 4 and Table 5 summarize the performances of CGAT with respect to different settings of M and L in the BRWS module. We can note that the best performance is achieved by setting M to 15. This indicates the most relevant entities in the non-local neighborhood of an entity can be captured by performing 15 times random walk sampling. Better performance can be achieved by setting L in the range between 4 and 12. Further increasing L causes more training time, however sometimes may cause the decrease in recommendation performances.

5.6 Parameter Sensitivity Study (RQ5)

We also conduct the experiments to analyze the impacts of the following hyper-parameters: the size of sampled neighbors S, the number of historical items N, the embedding dimension d, coefficient of KG loss λ_1, and the hyper-parameter γ of the biased random walk. Figure 5 summarizes the performances of CGAT with respect to different size of sampled neighbors S and the number of historical items N on different datasets. We can note that CGAT achieves the best performances on FM and ML datasets when S is set to 4, while larger S does not help further improve the performance. On the BC dataset, the best performances can be achieved by setting S to 32. In addition, we can also note that the optimal settings for N on FM, ML, and BC are 16, 16, and 32 respectively. Figure 6 shows the performance trends of CGAT with respect to different settings of d. From Figure 6, we can observe that better performances can usually be achieved by using a larger dimensionality of latent space. When d is large enough (*e.g.*, $d = 64$), further increasing d may not help improve the recommendation accuracy but increases the model complexity. Figure 7 shows the performance trend of CGAT with respect to different settings of λ_1. On FM and ML datasets, we can find that the performances achieved by setting λ_1 to 5×10^{-5} and 10^{-4} are better than that achieved by setting λ_1 to 0. This observation demonstrates that the KG structure constraint in Eq. (20) can help improve the recommendation accuracy. Moreover, we vary γ from 0.1 to 0.9. Figure 8 shows the performances of CGAT with respect to different settings of γ on FM, ML, and BC datasets. As shown in Figure 8, better performances are more likely to be achieved, when γ is smaller than 0.5. The optimal settings for γ on FM, ML, and BC datasets are 0.2, 0.3, and 0.2.

6 Conclusion and Future Work

This paper proposes a novel recommendation model, called Context-aware Graph Attention Network (CGAT). It explicitly exploits both local and non-local context information in KG and the interaction context information given by users' historical behaviors. Specifically, CGAT aggregates the local context information in KG by a user-specific graph attention mechanism, which captures users' personalized preferences on entities. To incorporate the non-local context in KG, a biased random walk based sampling process is used to extract important entities for the target entity over the item KG, and a GRU module is employed to explicitly aggregate these entity embeddings. Moreover, CGAT utilizes an item-specific attention mechanism to model the influences between items. The superiority of CGAT has been validated by comparing with state-of-the-art baselines on three real datasets. For future work, we would like to examine CGAT on more KG-based recommendation scenarios. We also intend to develop different aggregation strategies to integrate the context information in KG and interaction graph to improve the recommendation accuracy. Moreover, we are also interested in studying how to include the relation information in the non-local graph context. Another potential research direction is discovering more effective methods to select the high-order and important neighbors for entities.

7 Acknowledgments

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore. This research is also supported, in part, by the National Research Foundation, Prime Minister's Office, Singapore under its AI Singapore Programme (AISG Award No: AISG-GC-2019-003) and under its NRF Investigatorship Programme (NRFI Award No. NRF-NRFI05-2019-0002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore. This research is also supported, in part, by the National Natural Science Foundation of China (No. 61672481), and the Youth Innovation Promotion Association CAS (No. 2018495).

References

[1] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–45, 2014.

[2] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38, 2019.

[3] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *RecSys'10*, 2010, pp. 135–142.

[4] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *WWW'18*, 2018, pp. 1583–1592.

[5] Z. Sun, Q. Guo, J. Yang, H. Fang, G. Guo, J. Zhang, and R. Burke, "Research commentary on recommendations with side information: A survey and research directions," *Electronic Commerce Research and Applications*, vol. 37, p. 100879, 2019.

[6] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *AAAI'15*, 2015, pp. 2181–2187.

[7] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *WSDM'19*, 2019, pp. 105–113.

[8] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *KDD'16*. ACM, 2016, pp. 353–362.

[9] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *KDD'17*. ACM, 2017, pp. 635–644.

[10] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," in *KDD'18*, 2018, pp. 1531–1540.

[11] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, vol. 11, no. 9, p. 137, 2018.

[12] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *CIKM'18*. ACM, 2018, pp. 417–426.

[13] H. Wang, F. Zhang, X. Xie, and M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in *WWW'18*, 2018, pp. 1835–1844.

[14] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *WWW'19*. ACM, 2019, pp. 151–161.

[15] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang, "Knowledge graph convolutional networks for recommender systems with label smoothness regularization," in *KDD'19*, 2019, pp. 968–977.

[16] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *KDD'19*, 2019, pp. 950–958.

[17] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR'17*, 2017.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *ICLR'18*, 2018.

[20] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *SIGIR'19*, 2019, pp. 165–174.

[21] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *WWW'19*, 2019, pp. 3307–3313.

[22] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR'20*, 2020, pp. 639–648.

[23] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *KDD'08*. ACM, 2008, pp. 426–434.

[24] G. Piao and J. G. Breslin, "Transfer learning for item recommendations and knowledge graph completion in item related domains via a co-factorization model," in *European Semantic Web Conference*. Springer, 2018, pp. 496–511.

[25] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in *WWW'19*. ACM, 2019, pp. 2000–2010.

[26] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han, "Recommendation in heterogeneous information networks with implicit user feedback," in *RecSys'13*. ACM, 2013, pp. 347–350.

[27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI'09*. AUAI Press, 2009, pp. 452–461.

[28] Y. Xiao, R. Xiang, Y. Sun, Q. Gu, and J. Han, "Personalized entity recommendation: a heterogeneous information network approach," in *WSDM'14*, 2014, pp. 283–292.

[29] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *CIKM'15*. ACM, 2015, pp. 453–462.

[30] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD'17*, 2017, pp. 135–144.

[31] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 2018.

[32] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," in *RecSys'18*. ACM, 2018, pp. 297–305.

[33] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, and X. Ren, "Jointly learning explainable rules for recommendation with knowledge graph," in *WWW'19*. ACM, 2019, pp. 1210–1221.

[34] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *AAAI'19*, vol. 33, no. 01, 2019, pp. 5329–5336.

[35] X. Huang, Q. Fang, S. Qian, J. Sang, Y. Li, and C. Xu, "Explainable interaction-driven user modeling over knowledge graph for sequential recommendation," in *MM'19*, 2019, pp. 548–556.

[36] H. Chen, Y. Li, X. Sun, G. Xu, and H. Yin, "Temporal meta-path guided explainable recommendation," in *WSDM'21*, 2021, pp. 1056–1064.

[37] Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo, and Y. Zhang, "Reinforcement knowledge graph reasoning for explainable recommendation," in *SIGIR'19*, 2019, pp. 285–294.

[38] K. Zhao, X. Wang, Y. Zhang, L. Zhao, Z. Liu, C. Xing, and X. Xie, "Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs," in *SIGIR'20*, 2020, pp. 239–248.

[39] X. Wang, Y. Xu, X. He, Y. Cao, M. Wang, and T.-S. Chua, "Reinforced negative sampling over knowledge graph for recommendation," in *WWW'20*, 2020, pp. 99–109.

[40] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, 2020.

[41] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[42] C. Gallicchio and A. Micheli, "Graph echo state networks," in *IJCNN'10*, 2010, pp. 1–8.

[43] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[44] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *ICLR*, 2014.

[45] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, 2016, pp. 3844–3852.

[46] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.

[47] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 1993–2001.

[48] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS'17*, 2017, pp. 1025–1035.

[49] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," in *UAI'18*, 2018, pp. 339–349.

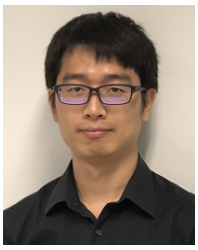
[50] C. Zhuang and Q. Ma, "Dual graph convolutional networks for graph-based semi-supervised classification," in *WWW'18*, 2018, p. 499–508.

[51] J. L. Keyulu Xu, Weihua Hu and S. Jegelka, "How powerful are graph neural networks?" in *ICLR'19*, 2019.

[52] H. Chen, H. Yin, T. Chen, W. Wang, X. Li, and X. Hu, "Social boosted recommendation with folded bipartite network embedding," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[53] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *AAAI'16*, 2016, pp. 1145–1152.

[54] D. Wang, C. Peng, and W. Zhu, "Structural deep network embedding," in *KDD'16*, 2016, pp. 1225–1234.
[55] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.
[56] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *IJCAI'18*, 2018, p. 2609–2615.
[57] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS'13*, 2013, pp. 2787–2795.
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[59] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender systems handbook*. Springer, 2011, pp. 257–297.
[60] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "Automatic multimedia cross-modal correlation discovery," in *KDD'04*, 2004, pp. 653–658.



Yong Liu is a Research Scientist at Alibaba-NTU Singapore Joint Research Institute and Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore. Before that, he was a Data Scientist at NTUC Enterprise, Singapore from November 2017 to July 2018, and a Research Scientist at Data Analytics Department, Institute for Infocomm Research (I2R), A*STAR, Singapore from November 2015 to October 2017. He received his Ph.D. from the School of Computer Science and Engineering at Nanyang Technological University in 2016 and B.S. from the Department of Electronic Science and Technology at University of Science and Technology of China in 2008. His research areas include various topics in machine learning and data mining. His research papers appear in leading international conferences and journals. He has been invited as a PC member of major conferences such as KDD, SIGIR, IJCAI, AAAI, CIKM, ICDM, and reviewer for IEEE/ACM transactions.

Susen Yang received the B.S. degree from the China University of Pertroleum, Tsingtao, China, in 2018. He is currently pursuing the M.S. degree from the University of Science and Technology of China, Hefei, China. His research interests include deep learning, recommendation system, and graph neural networks.

Yonghui Xu is currently a professor in the Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University. Before that, he was a research fellow in the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore. He received his Ph.D. from the School of Computer Science and Engineering at South China University of Technology in 2017 and BS from the Department of Mathematics and Information Science Engineering at Henan University of China in 2011. His research interests include machine learning, knowledge graphs, explainable AI, multi-modal data fusion, expert systems and their applications in e-commerce and healthcare. He has been invited as reviewer for top journals, such as, TKDE, TNNLS, IEEE Transactions on Cybernetics, Knowledge-Based System, Neurocomputing, and TKDD.

Chunyan Miao is a President's Chair Professor and the Chair of the School of Computer Science and Engineering (SCSE) at NTU Singapore. She received her PhD degree in Computer Engineering from NTU and was an NSERC Postdoctoral Fellow at Simon Fraser University (SFU), Canada. She was a founding faculty member of the Centre for Digital Media established by The University of British Columbia (UBC) and SFU. She was also a Tan Chin Tuan Engineering Fellow at Harvard and MIT. Dr. Miao has received over 20 Best Paper/innovation awards in Artificial intelligence (AI) and real world AI applications for her impactful research in health, ageing, education and smart services. She is a recipient of the prestigious NRF Investigatorship Award 2018. She also holds major research funding including MOH National Innovation Challenge (NIC) on Ageing award 2018 and NRF AI.SG Health Grand Challenge Award 2019. She is the Founding Director of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Singapore's first centre focusing on AI empowered solutions to population aging challenges. She is also the Founding Director of the Alibaba-NTU Singapore Joint Research Institute (JRI), Alibaba's first and largest JRI outside China. She is an Editor/Associate Editor of leading international journals including IJIT, IEEE Big Data, IEEE IoT, IEEE Access and IEEE Service Computing and has served as Chair/TPC member of international conferences such as IEEE ICA, ICAA, ACM KDD. She serves on various national committees, including the MOH City for All Ages and Health Tech, the IMDA TechSkills Accelerator (TeSA) and is the Chair of the SCS AI Ethics Review Committee. She was awarded a Public Administration Medal (Bronze) from the President of Singapore in 2016.

Min Wu is currently a Senior Research Scientist in the Machine Intellection Department at the Institute for Infocomm Research (I2R) under the Agency for Science, Technology and Research (A*STAR), Singapore. He received the B.Eng. from the University of Science and Technology of China (USTC), China in 2006 and his Ph.D. degree from Nanyang Technological University, Singapore in 2011. He received the best paper awards in the 15th International Conference on Bioinformatics (InCoB 2016) and the 20th International Conference on Database Systems for Advanced Applications (DASFAA 2015). He also won the IJCAI contest 2015 on repeated buyers prediction after sales promotion. His current research interests include machine learning, graph mining and bioinformatics.

Juyong Zhang is an associate professor in the School of Mathematical Sciences at University of Science and Technology of China. He received the BS degree from the University of Science and Technology of China in 2006, and the PhD degree from Nanyang Technological University, Singapore. His research interests include computer graphics, 3D computer vision, and numerical optimization. He is currently an AE for IEEE TMM and the Visual Computer.