Pairwise Intent Graph Embedding Learning for Context-Aware Recommendation

Dugang Liu
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University
Shenzhen, China
dugang.ldg@gmail.com

Yuhao Wu
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
wuyuhao2020@email.szu.edu.cn

Weixin Li
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, China
liweixin2021@email.szu.edu.cn

Xiaolian Zhang
Huawei 2012 Lab
Shenzhen, China
zhangxiaolian@huawei.com

Hao Wang
Huawei 2012 Lab
Shenzhen, China
wanghao225@huawei.com

Qinjuan Yang
Huawei 2012 Lab
Shenzhen, China
yangqinjuan@huawei.com

Zhong Ming*
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University
Shenzhen, China
mingz@szu.edu.cn

ABSTRACT

Although knowledge graph has shown their effectiveness in mitigating data sparsity in many recommendation tasks, they remain underutilized in context-aware recommender systems (CARS) with the specific sparsity challenges associated with the contextual features, i.e., feature sparsity and interaction sparsity. To bridge this gap, in this paper, we propose a novel pairwise intent graph embedding learning (PING) framework to efficiently integrate knowledge graphs into CARS. Specifically, our PING contains three modules: 1) a graph construction module is used to obtain a pairwise intent graph (PIG) containing nodes for users, items, entities, and enhanced intent, where enhanced intent nodes are generated by applying user intent fusion (UIF) on relational intent and contextual intent, and two sub-intents are derived from the semantic information and contextual information, respectively; 2) a pairwise intent joint graph convolution module is used to obtain the refined embeddings of all the features by executing a customized convolution strategy on PIG, where each enhanced intent node acts as a hub to efficiently propagate information among different features and between all the features and knowledge graph; 3) a recommendation module with the refined embeddings is used to replace the randomly initialized embeddings of downstream recommendation models to improve model performance. Finally, we conduct extensive experiments on three public datasets to verify the effectiveness and compatibility of our PING.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Context-aware recommendation, Graph embedding learning, Pairwise intent graph, User intent

ACM Reference Format:
Dugang Liu, Yuhao Wu, Weixin Li, Xiaolian Zhang, Hao Wang, Qinjuan Yang, and Zhong Ming. 2023. Pairwise Intent Graph Embedding Learning for Context-Aware Recommendation. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3604915.3608815

1 INTRODUCTION

Different from a traditional recommendation scenario, context-aware recommender systems (CARS) aim to integrate various contextual information to provide the users with more accurate fine-grained recommendations [2, 36]. The contextual information can be either explicitly observed or implicitly inferred [6, 22], and they can be integrated into multiple stages of the recommendation task, such as the pre-filtering, post-filtering, or modeling stages [1]. Among them, context-aware recommendation aimed at improving the modeling stage is a more popular paradigm and existing works can be divided into two main categories according to the underlying model architecture, including machine learning and neural network-based methods. The former aims to incorporate the contextual information into the model from the perspective

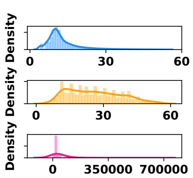
*Corresponding author

RecSys '23, September 18–22, 2023, Singapore, Singapore

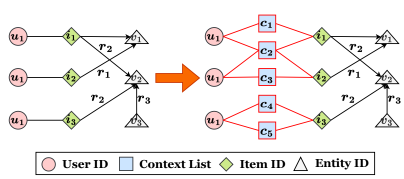
ACM ISBN 979-8-4007-0241-9/23/09…$15.00
https://doi.org/10.1145/3604915.3608815

of high-dimensional settings, and some classical machine learning methods can be adopted to solve it effectively, especially matrix factorization-, tensor factorization-, and factorization machine-based methods [4, 11, 19, 33, 37]. The latter utilizes some complex network architectures to more flexibly enhance and capture the relationships between the contextual features and the other features in different ways, such as attention mechanisms [17, 38], convolutional networks [9, 30] and graph learning techniques [5, 15, 16, 29].

Although existing methods for CARS have shown promising results, they still suffer from two specific sparsity challenges associated with the contextual features, i.e., feature sparsity and interaction sparsity [15]. Taking Amazon-Book used in our experiments as an example, Fig. 1(a) shows the interaction sparsity w.r.t the users and items, and feature sparsity, respectively, where the former means that the numbers of the associated contextual features are a long-tailed distribution w.r.t. the users (or items), and the latter means that most of the contextual features have a low frequency. Obviously, these two sparsity challenges make these methods prone to performance bottlenecks on inactive users, unpopular items, or uncommon contextual features. Given that knowledge graph (KG) is an effective solution to the data sparsity problem in many traditional recommendation tasks, an intuitive idea is to integrate it into CARS to alleviate the specific sparsity challenges mentioned above. However, a key difference is that these sparse challenges in CARS are closely related to the contextual features, and the contextual features play an essential role in CARS. As shown in Fig. 1(b), this means that the introduction of KG in CARS not only needs to enhance the information of users and items like the traditional recommendation but also needs to enhance the contextual information and the relationship between it and other information.



(a) Specific data sparsity in CARS



(b) Integrate KG into the traditional recommendation or CARS

Figure 1: (a) The distributions of the number of contextual features associated with each user (top row) and item (middle row), as well as the frequency statistics of the contextual features (bottom row), on Amazon-Book; and (b) The difference after integrating knowledge graph in the traditional recommendation and context-aware recommendation.

These unique properties make existing knowledge graph-based recommendation methods not easily migrated to CARS, and this also motivates us to design a more reasonable way to make CARS more effectively compatible with KG. In this paper, we propose a novel pairwise intent graph embedding learning (PING) framework to achieve this goal. First, a graph construction module is used to generate a novel pairwise intent graph (PIG) for CARS, where KG and the context features are used to construct some additional enhanced intent nodes through the proposed user intent fusion (UIF) strategy. Specifically, intent representations from two different perspectives are first learned, i.e., using the contextual information and the semantic information from KG to capture contextual intent and relational intent, respectively. Then, the two sub-intents are fused using the item as a hub connecting the contextual features and KG, which can well model the complex node relations after integrating KG in CARS. Second, a pairwise intent joint graph convolution module is used to exploit the above graph structure to obtain various refined embeddings, where each enhanced intent node acts as a hub to efficiently propagate information among different features and between all the features and KG. The obtained refined embeddings of all the features can later be integrated into an existing CARS model in a recommendation module to replace the original random initialization, thus improving its performance. Finally, we conduct extensive experiments on three public datasets to verify the effectiveness and compatibility of our PING.

2 RELATED WORK

In this section, we briefly review some relevant works on two research topics, including context-aware recommender systems and knowledge graph-based recommender systems.

2.1 Context-Aware Recommender Systems

Context-aware recommender systems (CARS) aim to introduce contextual information to provide the users with a fine-grained recommendation, where the contextual information can be either explicitly observed or implicitly inferred [1, 6, 22]. Typically, the contextual information can be used in the pre-filtering, post-filtering, or modeling stages of a recommendation task, where the last one is the more popular paradigm [1]. Depending on the model architecture employed, existing works in this line can be divided into two main categories, including machine learning and neural network-based methods. The former aims to extend a recommendation task to the high-dimensional settings to model various contextual information, and some representative works include matrix factorization, tensor factorization and factorization machines and their different variants, etc [4, 11, 19, 20, 33, 37]. The latter aims to more flexibly capture high-order and nonlinear relationships between different features by introducing some complex neural network structures, such as attention mechanisms [17, 38], convolution networks [9, 30], and graph learning techniques [5, 15, 16, 29]. This helps to learn better various feature information in CARS, especially the contextual features that play a crucial role. In addition, some work aims to combine the techniques of these two lines to synergistically produce better performance, such as NFM [10] and xDeepFM [13]. GCM [29] and UEG [15] are the two most related works to ours. GCM designs an attributed user-item bipartite graph for CARS, in which the contextual features are used as edge attributes between the users and items. Since the contextual features cannot benefit from the high-order propagation of information in GCM, UEG proposes a user-event graph structure, in which the intent driven by the contextual information is used as a central node, so that the contextual features can effectively participate in high-order propagation. Our PING is significantly different from theirs. In particular, we propose a pairwise intent graph structure, which introduces some enhanced intent nodes to better combine the contextual information with the semantic information in the knowledge graph and is beneficial to enhance the connection between different features further.

2.2 Knowledge Graph-Based Recommender Systems

As an auxiliary data source with rich semantic information, knowledge graph (KG) has been widely used in various recommendation tasks and demonstrated its effectiveness. Existing knowledge graph-based recommendation methods can be mainly divided into four categories, including embedding-based, path-based, propagation-based, and hyperbolic-based methods. Embedding-based methods directly obtain the entity and relation embeddings through various knowledge graph embedding (KGE) methods and introduce them into recommender systems in different ways [3, 23, 34]. Path-based methods aim to find semantic paths in KG, which are then used to construct latent high-order relations between the users and items. These paths can often be modeled by recurrent neural networks and attention mechanisms to predict user preferences [21, 27]. Propagation-based methods iteratively perform neighbor node-based information aggregation mechanisms and can discover high-order relations in an end-to-end manner. This is also the most popular line to integrate KG in recommendation systems, and many representative methods have emerged, such as CKAN [28], KGAT [25] and KGIN [26]. Hyperbolic-based methods aim to explore how to effectively use hyperbolic geometry and hyperbolic embeddings in recommender systems with KG to improve recommendation performance, which is an emerging direction [7]. Given the effectiveness of KG in addressing data sparsity in recommender systems [31], it is also intuitively beneficial for CARS tasks with two special issues, i.e., feature sparsity and interaction sparsity. However, although there are numerous knowledge graph-based recommendation methods, only a few works involve context-aware recommendation, and these works only consider preprocessing KG into the embeddings and then incorporate them into the models, and all the target a limited applicable scenario [18, 35]. Our PING aims to use an end-to-end manner to more effectively make KG beneficial to the learning of all the features in CARS, especially the contextual features, and it is easy to extend to different scenarios.

3 PRELIMINARIES

3.1 Problem Definition

In this subsection, we briefly give the definition and necessary notation of a context-aware recommendation task with the knowledge graph. We denote the sets of users, items, and contextual features in a typical contextual recommender system (CARS) as $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$, $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$ and $C = \{C^1, C^2, \ldots, C^R\}$, respectively. The common contextual features can be timestamps and locations, etc., and we do not consider the additional features of users and items for simplicity. Then, a user-item interaction instance can be represented as,

$$\mathbf{s}_k = \left[u^k, i^k, \mathbf{C}^k\right], \tag{1}$$

where $u^k \in \mathcal{U}$, $i^k \in \mathcal{I}$ and $\mathbf{C}^k \subset C$ denote the user, item, and context involved in the k-th instance, respectively. Furthermore, we assume that a knowledge graph $\mathcal{G}_{kg} = \{(h, r, t) \,|h, t \in \mathcal{V}, r \in \mathcal{R}\}$ associated with the items is available, where each triple (h, r, t) indicates that a relation r exists from head entity h to tail entity t. For example, $(Tom\ Cruise, star, Top\ Gun)$ describes that *Tom Cruise* is the *star* of movie *Top Gun*. The goal of context-aware recommendation with the knowledge graph is to use a knowledge graph as an auxiliary data source and accurately predict an item i that is most likely to be interacted by a user u under a context $\mathbf{C}$. Obviously, accurately learning and integrating contextual information is crucial for CARS. However, as mentioned earlier, two specific data sparsity problems, i.e., feature sparsity and interaction sparsity, pose challenges for this.

3.2 Base Model

In this work, we focus on utilizing a new graph structure and graph embedding learning to improve the representation of all the features in CARS, which can replace randomly initialized representations and enable downstream recommendation models to effectively alleviate the data sparsity problem. Therefore, following previous work [4, 8, 30], we take the factorization machine as an example of the downstream recommendation model and refer to it as the base model in this paper. Note that in order to verify the compatibility of our framework, we will also analyze the performance of our framework with other types of downstream recommendation models in the experiments.

3.2.1 Initial Embedding. In practice, an instance $\mathbf{s}_k$ in context-aware recommendation is usually represented in a sparse high-dimensional binary form, which may be one-hot vectors or multi-hot vectors. By applying the corresponding embedding layer to each feature in $\mathbf{s}_k$, where the embedding layer contains an embedding table associated with the feature values, we can obtain their dense low-dimensional real-valued form. Finally, we concatenate these embedding representation vectors to get the representation of the entire instance,

$$\mathbf{E}_{\mathbf{s}_k} = [\mathbf{e}_{u^k}, \mathbf{e}_{i^k}, \mathbf{e}_{\mathbf{C}^k}]. \tag{2}$$

3.2.2 Feature Interaction. After taking the embedding representation of an instance as input, the factorization machine uses a feature interaction layer involving second-order interactions to capture the user preferences,

$$\hat{y}(\mathbf{s}_k) = \sigma(b_g + \sum b_\star + \frac{1}{2}[(\sum \mathbf{e}_\star)^2 - \sum \mathbf{e}_\star^\top \mathbf{e}_\star]), \tag{3}$$

where $\star \in \{u^k, i^k, \mathbf{C}^k\}$, b_g is the global bias, $b_\star$ is the feature bias term, and $\sigma(\cdot)$ is the sigmoid activation function.

3.2.3 Model Training. We use the point-wise log loss as the objective function in our experiments,

$$\mathcal{L} = -\frac{1}{|\mathcal{S}'|} \sum_{(\mathbf{s}_i, y) \in \mathcal{S}'} y_i \log \hat{y}(\mathbf{s}_i) + (1 - y_i) \log(1 - \hat{y}(\mathbf{s}_i)), \tag{4}$$

where $\mathcal{S}$ is the set of interaction instances, $\mathcal{S}' = \mathcal{S} \cup \mathcal{S}^-$, and $\mathcal{S}^-$ is a set of negative instances randomly selected for each positive instance in $\mathcal{S}$ from a candidate set of items that the corresponding user has not interacted with under the same context.

4 PAIRWISE INTENT GRAPH EMBEDDING LEARNING

As mentioned in Sec. 2, knowledge graphs have been shown to help alleviate data sparsity in recommender systems. However, most of the existing works for context-aware recommendation rarely involve knowledge graphs, and systematic guidance to effectively utilize knowledge graphs is still lacking. To bridge this gap, we propose a pairwise intent graph (PIG) to effectively integrate the semantic information of knowledge graphs with the user, item, and contextual information, where sub-intents driven by their respective information will be fused into the enhanced intent node to enhance information propagation. We coin the framework as pairwise intent graph embedding learning (PING). Our PING consists of three modules, including graph construction, pairwise intent joint graph convolution, and downstream recommendation with the refined embedding vectors. Next, we will introduce each module in detail according to the training pipeline. We illustrate the architecture of our proposed framework in Fig. 2.

4.1 Graph Construction

4.1.1 Personal Graph Combined with Knowledge Graph. All the historical interaction behaviors of a user in CARS can usually be organized into a form of personal graph $\mathcal{G}_{pg} = \langle \mathcal{V}_{pg}, \mathcal{E}_{pg} \rangle$ [32], in which the user node representing oneself is used as a center. As shown on the left side of Fig. 2, the nodes include the user ID and the interacted items, and the edges include interactions between the users and the items, and temporal relationships between the items, i.e., $\mathcal{V}_{pg} = \{u\} \cup \mathcal{I}$ and $\mathcal{E}_{pg} = \mathcal{E}_{ui} \cup \mathcal{E}_{ii}$. Note that a list of contextual features is used as edge features of $\mathcal{E}_{ui}$. After the corresponding mapping between items and entities, the personal graph can further integrate KG. Formally, we have $\mathcal{G}_{pg\&kg} = \langle \mathcal{V}_{pg\&kg}, \mathcal{E}_{pg\&kg} \rangle$, $\mathcal{V}_{pg\&kg} = \{u\} \cup \mathcal{I} \cup \mathcal{V}$ and $\mathcal{E}_{pg\&kg} = \mathcal{E}_{ui} \cup \mathcal{E}_{ii} \cup \mathcal{E}_{iv} \cup \mathcal{E}_{vv}$. Similarly, a relation r can be viewed as an edge attribute on $\mathcal{E}_{iv}$ and $\mathcal{E}_{vv}$. Obviously, it is very difficult to apply this straightforward graph structure in CARS. On one hand, the existence of many heterogeneous nodes means that an unreasonable convolution method will bring too much noise in the information propagation. On the other hand, the entities are weakly connected to the user and contextual information, which may also weaken the extent to which such information can benefit from KG.

4.1.2 Pairwise Intent Graph. To address the above problems, we propose a new graph structure called pairwise intent graph (PIG) for the context-aware recommendation. Specifically, to construct the pairwise intent graph, we propose user intent fusion (UIF) to capture the fine-grained user intent in each instance and generate corresponding enhanced intent nodes $\mathcal{T} = \{\bar{t}^1, \bar{t}^2, \ldots, \bar{t}^k, \ldots\}$. First, we model a sub-intent (i.e., relational intent) driven by the semantic information of KG. The idea behind this process is that the users usually interact with the items because of some relations between the entities, for example, a user may give feedback because he likes the star or director of a movie. Inspired by previous works [3, 26], we define a set of shared intents $\mathcal{P}$ derived from the set of relations in KG, one of which is computed as follows, i.e., we assume that there is a candidate set of $|\mathcal{P}|$ relational intents from which each user can find their own.

$$\mathbf{e}_p = \sum_{r \in \mathcal{R}} \alpha_{rp} \mathbf{e}_r, \tag{5}$$

$$\alpha_{rp} = \frac{\exp(w_{rp})}{\sum_{r' \in \mathcal{R}} \exp(w_{r'p})}, \tag{6}$$

where $p \in \mathcal{P}$, $\mathbf{e}_r$ is the embedding of a relation r, and w_{rp} is a trainable weight indicating the degree of association between a certain relation r and a certain shared intent p. Since shared intent is global, we further introduce user information to obtain personalized relational intent t'_u as follows. We refer to the above process as the relational intent attention generation, and an illustration of it can be found on the left side of Fig. 2.

$$\mathbf{e}_{t'_u} = \sum_{p \in \mathcal{P}} \beta_{up} \mathbf{e}_p, \tag{7}$$

$$\beta_{up} = \frac{\exp(\mathbf{e}_p^\top \mathbf{e}_u)}{\sum_{p' \in \mathcal{P}} \exp(\mathbf{e}_{p'}^\top \mathbf{e}_u)}. \tag{8}$$

Second, we model a sub-intent (i.e., contextual intent) driven by contextual information. Assume that the number of context features associated with each instance is defined as Z, i.e., $\mathbf{C}^k = \{c_1^k, c_2^k, \ldots, c_Z^k\}$. Inspired by previous works [15], since a user's behavior may also be influenced by the preceding behaviors rather than the context alone, we additionally introduce a context-specific feature c_{Z+1}^k to represent the last interacted item before the current instance. Then, a user's contextual intent on an instance is obtained as follows. The idea behind this process is to capture a subset of contextual features that a user pays more attention to in an interaction.

$$\gamma_z^k = \text{Softmax}(\mathbf{W}_0^\top \text{Relu}(\mathbf{W}_1 \mathbf{e}_{u^k} + \mathbf{W}_2 \mathbf{e}_{c_z^k} + \mathbf{b}_1)), \tag{9}$$

$$\mathbf{e}_{t^k} = \sum_{z=1}^{Z+1} \gamma_z^k \mathbf{e}_{c_z^k}, \tag{10}$$

where $\mathbf{W}_0 \in \mathbb{R}^{d \times 1}, \mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}, \mathbf{b}_1 \in \mathbb{R}^{d \times 1}$ are trainable parameters, d is the embedding size, and $\mathbf{e}_{t^k}$ is the embedding representation of the intent node corresponding to the k-th instance. We refer to the above process as the contextual intent attention mechanism, and an illustration of it can be found on the right side of Fig. 2. Finally, we fuse the two sub-intents to obtain a more fine-grained user intent and introduce it as an enhanced intent node into the proposed pairwise intent graph. Here we use the mean fusion for simplicity, i.e.,

$$\mathbf{e}_{\bar{t}^k} = \frac{(\mathbf{e}_{t'_u} + \mathbf{e}_{t^k})}{2}. \tag{11}$$

The idea behind the proposed user intent fusion is that sub-intents obtained by independent modeling avoid the introduction of noise, and the fusion of sub-intents enhances the association between semantic information and user and contextual information.

4.2 Pairwise Intent Joint Graph Convolution

Clearly, existing graph embedding learning techniques are unsuitable for our pairwise intent graph due to the introduction of KG and enhanced intent nodes. Therefore, in this section, we introduce the proposed pairwise intent joint graph convolution to utilize the

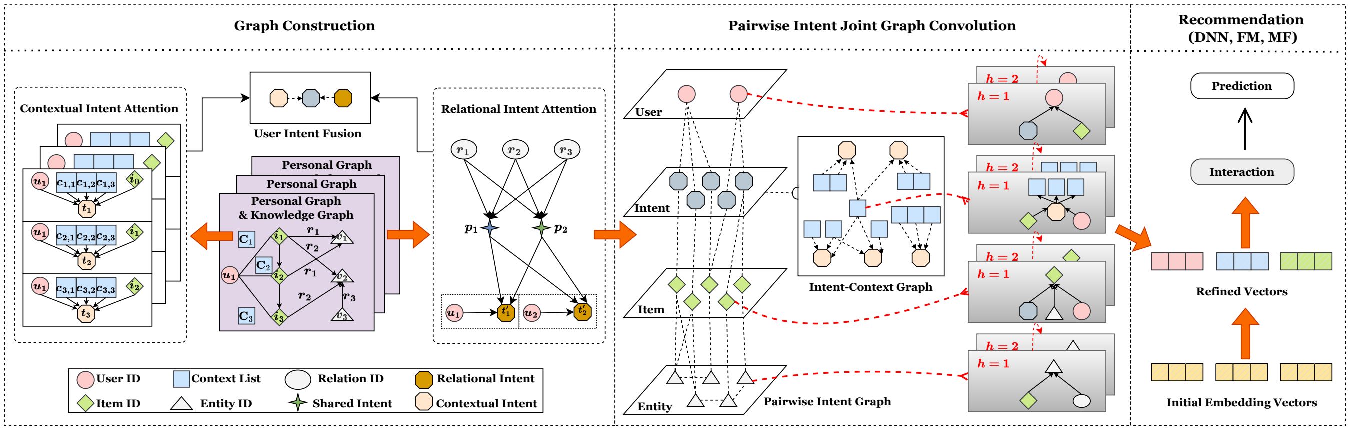


Figure 2: The architecture of our PING framework consists of three modules: 1) the graph construction module is used to construct the pairwise intent graph, in which user intent fusion is used to obtain the required sub-intents from knowledge graph and contextual information, respectively, and fuse them to form the enhanced intent nodes; 2) the pairwise intent joint graph convolution module is used to exploit the above graph structure to obtain various refined embeddings; and 3) the recommendation module uses the refined feature embeddings to improve the performance of a downstream recommendation model.

newly proposed graph structure fully. Specifically, our goal is to effectively use KG to enhance the high-order information synergy among the users, items, and contexts through the constructed enhanced intent nodes, thereby further alleviating the data sparsity problem and improving the recommendation performance of CARS. For ease of understanding, the information propagation for different nodes in pairwise intent joint graph convolution is illustrated in Fig. 3.

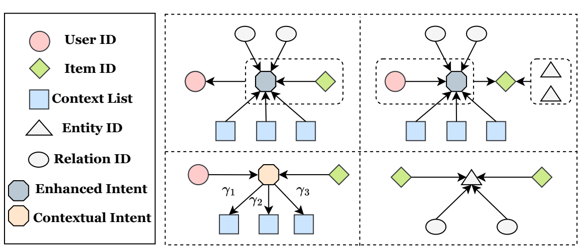


Figure 3: An illustration of information propagation for the nodes of users, items, contextual features, and entities in pairwise intent joint graph convolution.

4.2.1 Information Propagation for the Users. For a user, we expect it can effectively capture the item and contextual information with the assistance of knowledge graphs. Therefore, as shown in the upper left corner of Fig. 3, we use an enhanced intent node as the hub to complete the desired information propagation,

$$\mathbf{p}_{u^k,k}^{(h)} = \mathbf{p}_{i^k}^{(h-1)} + \mathbf{p}_{\bar{t}^k}^{(h-1)}, \tag{12}$$

where $\mathbf{p}_{u^k,k}^{(h)}$ is the information representation passed to the user associated with the k-th instance in layer h of graph convolution, $\mathbf{p}_{i^k}^{(h-1)}$ and $\mathbf{p}_{\bar{t}^k}^{(h-1)}$ are the item embedding and enhanced intent node embedding at layer $h-1$, respectively, and $\mathbf{p}_{i^k}^{(0)} = \mathbf{e}_{i^k}, \mathbf{p}_{\bar{t}^k}^{(0)} = \mathbf{e}_{\bar{t}^k}$. We then use an aggregation function to obtain each user's embedding at layer h,

$$\mathbf{p}_u^{(h)} = \frac{1}{\sqrt{|\{k|u^k = u\}|}} \sum_{k,u^k=u} \mathbf{p}_{u,k}^{(h)}. \tag{13}$$

Finally, we can obtain a refined user embedding by averaging the embeddings of each layer,

$$\hat{\mathbf{p}}_u = \frac{1}{H+1} \sum_{h=0}^{H} \mathbf{p}_u^{(h)}. \tag{14}$$

4.2.2 Information Propagation for the Items. Similarly, as shown in the upper right corner of Fig. 3, we use an enhanced intent node as the hub with the assistance of knowledge graphs to enable items to effectively capture the user and contextual information,

$$\mathbf{p}_{i^k,k}^{(h)} = \mathbf{p}_{u^k}^{(h-1)} + \mathbf{p}_{\bar{t}^k}^{(h-1)}, \tag{15}$$

where $\mathbf{p}_{u^k}^{(h-1)}$ is the user embedding at layer $h-1$. The embedding of an item i at layer h can be obtained by using an aggregation function and receiving additional information,

$$\mathbf{p}_i^{(h)} = \frac{1}{\sqrt{|\{k|i^k = i\}|}} \sum_{k,i^k=i} \mathbf{p}_{i,k}^{(h)} + \frac{1}{|\mathcal{N}_i|} \sum_{(r,v)\in\mathcal{N}_i} \mathbf{e}_r \odot \mathbf{p}_v^{(h-1)}, \tag{16}$$

where $\mathcal{N}_i$ denotes the set of relations and entities associated with item i, $\mathbf{p}_v^{(h-1)}$ is the entity embedding at layer $h-1$ and $\mathbf{p}_v^{(0)} = \mathbf{e}_v$. We then obtain a refined item embedding as follows,

$$\hat{\mathbf{p}}_i = \frac{1}{H+1} \sum_{h=0}^{H} \mathbf{p}_i^{(h)}. \tag{17}$$

4.2.3 *Information Propagation for the Context.*

For the contextual features, as shown in the lower left corner of Fig. 3, the user and item information will be propagated to intent nodes first, and then different contextual features will distribute the information according to the contextual attention distribution,

$$\mathbf{p}_{t^k,k}^{(h)} = \mathbf{p}_{u^k}^{(h-1)} + \mathbf{p}_{i^k}^{(h-1)}. \tag{18}$$

$$\mathbf{p}_{c_z^k,k}^{(h)} = \gamma_z^k \mathbf{p}_{t^k,k}^{(h)}. \tag{19}$$

Finally, we can get the embedding of the contextual features at layer h and the final refined embedding as follows,

$$\mathbf{p}_{c_z}^{(h)} = \frac{1}{\sqrt{|\{k|c_z^k = c_z\}|}} \sum_{k,c_z^k=c_z} \mathbf{p}_{c_z^k,k}^{(h)}. \tag{20}$$

$$\hat{\mathbf{p}}_{c_z} = \frac{1}{H+1} \sum_{h=0}^{H} \mathbf{p}_{c_z}^{(h)}. \tag{21}$$

Note that since the contextual features are only related to contextual intent, information propagation only uses sub-intent nodes obtained by contextual intent attention instead of enhanced intent nodes. In addition, since the calculation of Eq.(12) and Eq.(15) in the next layer requires contextual intent at layer h, it needs to feed the information back to the sub-intent nodes,

$$\mathbf{p}_{t^k}^{(h)} = \sum \gamma_z^k \mathbf{p}_{c_z^k}^{(h)}. \tag{22}$$

4.2.4 *Information Propagation for the Entity.*

As shown in the lower right corner of Fig. 3, the information propagation of entities only involves the items and relations, which can be expressed as,

$$\hat{\mathbf{p}}_{v}^{(h)} = \frac{1}{|\mathcal{N}_v|} \sum_{(r,i)\in\mathcal{N}_v} \mathbf{e}_r \odot \mathbf{p}_{i}^{(h-1)}, \tag{23}$$

where $\mathcal{N}_v$ denotes the set of relations and items associated with entity v. Comparing with Eq.(2), after performing pairwise intent joint graph convolution, we can obtain a set of corresponding refined embeddings, and can integrate them into arbitrary CARS models in place of randomly initialized embeddings.

$$\mathbf{P}_{\mathbf{s}_k} = [\hat{\mathbf{p}}_{u^k}, \hat{\mathbf{p}}_{i^k}, \hat{\mathbf{p}}_{\mathbf{C}^k}]. \tag{24}$$

4.3 Complexity Analysis

In this subsection, we analyze the time complexity of our PING. Since the refined embeddings can be obtained in offline training and directly used for online inference, the tolerance for time complexity will be greater. The time complexity of our PING in this case is the same as that of the base model. For model training, the main cost of our PING is on the pairwise intent joint graph convolution, and its complexity is $O(Z \cdot |\mathcal{G}_{pig}| \cdot d)$, where $|\mathcal{G}_{pig}|$ denotes the number of edges in $\mathcal{G}_{pig}$. Compared to representative graph-based context-aware methods, the additional complexity is linear with the number of connections between entities, relations, and items. Typically, the number of these connections is much smaller than the connections between users, items, and contexts, which means that our PING has comparable complexity to the baselines.

5 EMPIRICAL EVALUATIONS

In this section, we conduct experiments with the aim of answering the following six key questions. Note that the source codes are available at https://github.com/dgliu/RecSys23_PING.

- **RQ1**: How does our PING perform compared to the baselines?
- **RQ2**: What is the role of each module in our PING?
- **RQ3**: What are the characteristics of intent attention obtained in our PING?
- **RQ4**: How does our PING perform on the two challenges of CARS, i.e., interaction sparsity and feature sparsity?
- **RQ5**: How is the compatibility of our PING?
- **RQ6**: What is the impact of the hyperparameters in our PING on performance?

5.1 Experimental Setup

5.1.1 *Datasets.*

To evaluate the effectiveness of the proposed framework, we need to use public datasets with KG in our experiments. Following the settings of previous works [7, 25, 26], we conduct experiments on three public datasets including Amazon-book[1], Yelp2018[2] and Last-FM[3].

- Amazon-book is a subset corresponding to the book product category in the Amazon dataset. In this dataset, a large number of ratings and comments from the users on book products are provided, and each book product can be linked to an external knowledge graph to obtain additional knowledge.
- Yelp2018 is the 2018 edition of the Yelp Challenge dataset. It provides extensive records of user interactions with local businesses, such as restaurants and bars, in metropolitan areas of different countries. Furthermore, additional item knowledge can be extracted as KG data, such as category, location, and attribute, through the attached local business information network.
- Last-FM is a music listening dataset collected from the Last.fm online music platform, where each interaction can be expressed as $(user\_id, artist\_id, album\_id, track\_id, timestamp)$. Here we view the tracks as the items. Similarly, by mapping each track with an external knowledge graph, we can obtain their corresponding additional item knowledge.

5.1.2 *Dataset Preprocessing.*

For the acquisition of the item KG corresponding to each dataset, we follow previous work [25] and use the open-source data they provide[4]. Please refer to the original paper for more information on the knowledge-mapping process. Similarly, to construct the corresponding contextual features for each dataset, we follow the setting of previous work [15]. Specifically, for the Yelp2018 dataset, there are four kinds of context, i.e., $city$, $year$, $month$, and day_of_week (DoW). In the Amazon-book dataset and the Last-FM dataset, the contextual features include $year$, $month$, day, and day_of_week. We use the same data partition as in previous studies [25, 26], i.e., 70% for training, 10% for validation, and the rest 20% for test. We summarize the statistics of the three processed datasets in Table 1.

[1] http://jmcauley.ucsd.edu/data/amazon/
[2] https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset/versions/8
[3] http://www.cp.jku.at/datasets/LFM-1b/
[4] https://github.com/xiangwang1223/knowledge_graph_attention_network

Table 1: Statistics of the processed datasets.

		Amazon-book	Yelp2018	Last-FM
User-Item Interaction	#Users	70,679	23,566	45,919
	#Items	24,915	48,123	45,538
	#Interactions	847,733	3,034,796	1,185,068
	#Contextual Feature	59	535	133
Knowledge Graph	#Entities	88,572	58,266	90,961
	#Relations	39	9	42
	#Triplets	2,557,746	464,567	1,853,704

5.1.3 Baselines. We select the representative methods from the two related research topics summarized in Sec. 2, including context-aware methods (FM, GCM, and UEG), knowledge graph (KG)-based methods (CKE, KGNN-LS, CKAN, KGAT, KGIN, and HAKG), and KG-free method (MF).

- **MF** [12]: This is one of the most classical recommendation methods, which only considers user-item interactions without knowledge graph. Here, we use ID embeddings of users and items to perform the prediction.
- **FM** [19]: This is one of the most classical context-aware methods, which considers the second-order feature interactions among input features. Here, We consider a user ID, an item ID, and the contextual features as input features.
- **CKE** [34]: This is an embedding-based knowledge graph utilization method, which uses semantic embeddings derived from TransR [14] to enhance the MF framework.
- **KGNN-LS** [24]: This is a propagation-based knowledge graph utilization method, which converts knowledge graphs into user-specific graphs, and then introduces user preferences and label smoothness in the information aggregation stage to generate personalized item embeddings.
- **CKAN** [28]: This method is based on KGNN-LS and introduces different aggregation schemes on the user-item graph and knowledge graphs to better encode the embeddings.
- **KGAT** [25]: This is a propagation-based knowledge graph utilization method, which combines the user-item graph and knowledge graphs and introduces a unified attention aggregation mechanism to obtain better user and item embeddings.
- **KGIN** [26]: This is a state-of-the-art propagation-based knowledge graph utilization method, which models the intents behind user behaviors and designs a new relation-aware information aggregation mechanism to capture long-range connectivity in knowledge graphs.
- **HAKG** [7]: This is a state-of-the-art hyperbolic-based knowledge graph utilization method, which designs a hyperbolic aggregation scheme to collect the relational context over knowledge graphs and introduces an angle constraint and a dual item embeddings design to better capture the high-order collaborative signals of the items.
- **GCM** [29]: This is a state-of-the-art context-aware method, which proposes an attributed user-item bipartite graph that treats contextual features as edge attributes and designs a new convolution method to fully explore the information of this graph structure.
- **UEG** [15]: This is a state-of-the-art context-aware method, which models the user's intentions for contextual features to generate a user-event graph structure, and proposes a corresponding convolution method to propagate information between users, items, and contexts effectively.

5.1.4 Evaluation Metrics. We evaluate the recommendation performance via two widely used ranking-oriented metrics, i.e., recall (R@k) and normalized discounted cumulative gain (N@k). We report the average metrics for all users in the testing set, where k is set to 20 [7, 25, 26]. The candidate items to be recommended for a user are from the set of items that have not been interacted with by the user.

5.1.5 Implementation Details. We implement our PING in TensorFlow 1.15. For the adopted baselines, we use the open-source implementations and parameter settings provided by previous studies [7, 15, 25, 26, 29], where the embedding size is set to 64, the batch size is set to 1024, Adam is used as the optimizer, and the learning rates for the context-aware baselines and the knowledge graph-based baselines are set to 0.001 and 0.0001, respectively. For our method, our search scope includes the number of GNN layers H in the range of $\{1, 2, 3\}$, the regularization weight γ in the range of $\{1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}\}$, and the number of relational intents $|\mathcal{P}|$ in the range of $\{1, 2, 4, 8\}$. The other parameters remained the same as the baselines. We perform a grid search to tune the hyper-parameters by evaluating the summation of R@20 and N@20. In addition, we also adopted an early stopping strategy with the patience set to 10 times to avoid over-fitting to the training set.

5.2 RQ1: Performance Comparison

We report the comparison results in Table 2. Note that to further illustrate the key role of contextual features in the context-aware recommendation, we also evaluate the performance of context-aware baselines without using contextual features for prediction i.e., GCM (w/o c) and UEG (w/o c). From the results in Table 2, we can have the following observations: 1) Knowledge graph-based baselines can benefit from the semantic information provided by knowledge graphs to better model user and item embeddings, and achieve better performance than MF; 2) Context-aware baselines have better performance than the above two, which indicates the crucial role of contextual features in CARS. Conversely, when contextual features are not available at the prediction stage, the performance of these methods suffers greatly, i.e., GCM (w/o c) and UEG (w/o c); 3) UEG performs the best among the baselines by modeling the intent behind contextual features and constructing a user-event graph with the intent nodes as the hub connecting users, items, and contexts. This means that effectively capturing the intent behind user interactions and considering a reasonable graph structure adapted to CARS can contribute a better performance; and 4) Our PING consistently outperforms all the baselines. This demonstrates the effectiveness of the proposed user intent fusion (UIF) and pairwise intent graph. In particular, our PING inherits and improves all the above beneficial observations. Specifically, UIF can combine knowledge graphs to capture more fine-grained user intent, and the proposed pairwise intent graph can rationally and effectively utilize knowledge graphs to further enhance the learning of user, item, and contextual embeddings.

Table 2: Results on all datasets, where the best and second best results are marked in bold and underlined, respectively. Note that * indicates a significance level of $p \leq 0.05$ based on a two-sample t-test between our method and the best baseline.

Dataset	Amazon-book		Yelp2018		Last-FM	
Metrics	R@20	N@20	R@20	N@20	R@20	N@20
MF	0.1300	0.0678	0.0627	0.0413	0.0724	0.0617
CKE	0.1342	0.0698	0.0653	0.0423	0.0732	0.0630
KGNN-LS	0.1362	0.0560	0.0671	0.0422	0.0880	0.0642
CKAN	0.1442	0.0698	0.0646	0.0441	0.0812	0.0660
KGAT	0.1487	0.0799	0.0705	0.0463	0.0873	0.0744
KGIN	0.1687	0.0915	0.0698	0.0451	0.0978	0.0848
HAKG	0.1421	0.0863	0.0778	0.0501	0.1008	<u>0.0931</u>
FM	0.1565	0.0793	0.1186	0.0550	0.1084	0.0433
GCM (w/o c)	0.1371	0.0618	0.0761	0.0309	0.0839	0.0387
GCM	0.1863	0.0919	0.1285	0.0556	<u>0.1088</u>	0.0496
UEG (w/o c)	0.1399	0.0623	0.0765	0.0317	0.0747	0.0346
UEG	<u>0.2872</u>	<u>0.1583</u>	<u>0.1352</u>	<u>0.0613</u>	0.0940	0.0441
PING	**0.3043***	**0.1759***	**0.2132***	**0.1336***	**0.1752***	**0.1014***

5.3 RQ2: Ablation Study of PING

To analyze the contribution of user intent fusion (UIF) and each sub-intent in our PING, we conduct an ablation study and report the results in Table 3. We evaluate the performance of our PING when user intent fusion (UIF) excludes relational intent (denoted as 'w/o RI'), excludes contextual intent (denoted as 'w/o CI'), and excludes the both (denoted 'w/o RI&CI'), respectively. From the results in Table 3, we have the following observations: 1) **'w/o RI&CI' vs. 'w/o RI', 'w/o CI'**. The variant that removes either sub-intent in user intent fusion beats the variant that removes both sub-intents. This means that the introduction of user intent is beneficial for the context-aware recommendation, whether relational intent driven by knowledge graphs or contextual intent driven by contextual features, and this is also consistent with the observations in Table 2. 2) **'w/o RI' vs. 'w/o CI'**. The variant removing relational intent outperforms the variant removing contextual intent in user intent fusion, indicating that contextual intent brings more gain than relational intent in the context-aware recommendation. This is expected, since the learning of contextual features plays a key role in contextual recommendation, and contextual intent is closely related to them. 3) **PING vs. 'w/o RI', 'w/o CI'**. Our PING achieves the best performance, which demonstrates the effectiveness of the proposed user intent fusion. In particular, the proposed user intent fusion can make the two sub-intents synergistic, and enable our PING to further enhance the high-order information propagation and collaboration between the users, items, and context nodes through enhanced intent nodes.

Table 3: Results of the ablation studies on all datasets, where the best results are marked in bold.

Dataset	Amazon-book		Yelp2018		Last-FM	
Metrics	R@20	N@20	R@20	N@20	R@20	N@20
PING	**0.3043**	**0.1759**	**0.2132**	**0.1336**	**0.1752**	**0.1014**
w/o RI	0.2714	0.1546	0.1677	0.0886	0.1459	0.0731
w/o CI	0.2537	0.1254	0.1521	0.0715	0.1439	0.0673
w/o RI&CI	0.1863	0.0919	0.1285	0.0556	0.1088	0.0496

5.4 RQ3&RQ4: In-depth Analysis of PING

Since relational intent and contextual intent are potentially beneficial to improve the interpretability of the framework, we are curious about the characteristics of the different attention weights obtained by our PING. We show the relational intent and contextual intent corresponding to two instances on Amazon-book and Yelp2018 in Fig. 4, respectively. We can observe that contextual intent captures the focus of different users, i.e. the user 16591 on Amazon-book is susceptible to the previous interaction, and the user 114 on Yelp2018 focuses more on location. We can also observe similar results on relational intentions, for example, the user 114 on Yelp2018 is more concerned with aspects that reflect the quality of the restaurant, such as table service and star rating.

To analyze whether our PING can effectively alleviate the two problems of feature sparsity and interaction sparsity, we conduct three studies about GCM, UEG, and our PING on Amazon-book. Specifically, we first count the number of contextual features associated with each user and each item, and the frequency of each contextual feature, respectively. Then, according to the obtained statistical results, we group all users, items, and contexts, respectively, and calculate the average result within each group in turn. The results are shown in Fig. 5(a), 5(b) and 5(c). We can observe that our PING has a significant improvement in all groups. The above results show that our PING can not only effectively alleviate these two key challenges in CARS, but also further enhance the learning of embeddings for the remaining groups of users, items, and contexts.

5.5 RQ5: Compatibility Evaluation of PING

To verify the compatibility of our PING for various downstream recommendation models, we evaluate the performance of our PING combined with three typical recommendation models, i.e., MF, FM, and MLP. For a more comprehensive comparison, we also evaluate the compatibility of the two most relevant context-aware baselines, i.e., GCM and UEG. The results are shown in Fig. 6. We can find that our PING consistently outperforms the base downstream recommendation model (i.e., using randomized initial embedding vectors instead of refined ones), GCM, and UEG in all the cases. This shows that in practice, our PING can be used as a general preprocessing framework to provide the refined embedding vectors for different downstream recommendation models, thereby improving their recommendation performance. Compared with GCM and UEG, the proposed pairwise intent graph can effectively combine knowledge graph and contextual features to form pair-aware intent nodes for capturing more fine-grained user intent. This enables the proposed joint graph convolution based on this graph structure to obtain the refined embedding vectors that are more beneficial to downstream recommendation models. In particular, UEG only models coarse-grained user intent based on contextual features, which cannot maintain good performance in some cases (such as on Last-FM), and our PING has a steady performance improvement in all the cases. An interesting observation is that FM as a downstream recommendation model has a more stable performance in most cases, which may be an important reason why FM and its variants are so popular in CARS. Furthermore, this advantage is further amplified

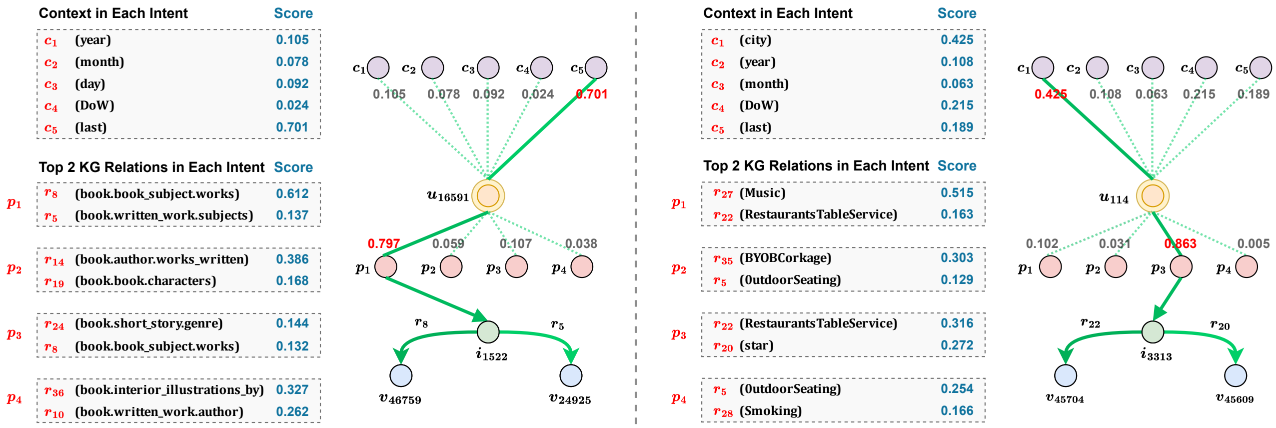
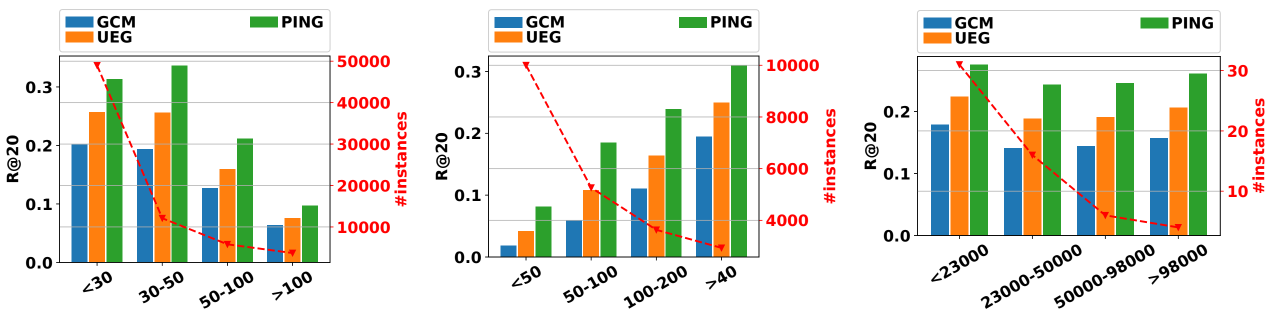


Figure 4: Explanations of user intents and real cases in Amazon-Book (left) and Yelp2018 (right). Best viewed in color.



(a) u-c interaction (b) i-c interaction (c) context frequency

Figure 5: Recommendation performance of our PING with different user-context (u-c) interaction levels, item-context (i-c) interaction levels, and context frequencies on Amazon-book.

after obtaining refined features with better expressive power for all the features.

5.6 RQ6: Parameter Sensitivity Analysis of PING

In this section, we conduct experiments on some key hyperparameters to analyze the results of our PING under different parameter values, including the number of convolutional layers, regularization weight, and the number of shared intent.

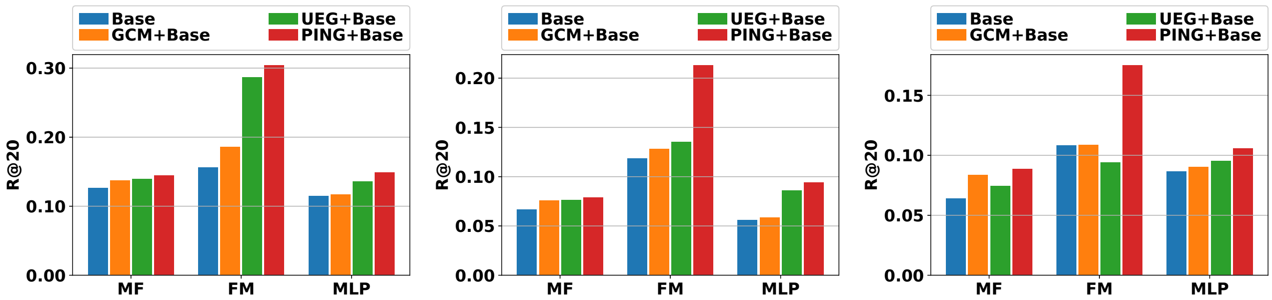
5.6.1 Impact of the Number of Layers. Since the number of convolutional layers directly affects the high-order information propagation between the users, items, and context nodes, it obviously has an impact on the performance of our PING. We consider the cases where the number of layers is set to 1, 2, and 3 respectively, and report the corresponding results in Table 4. As shown in Table 4, when the number of layers is 2, our PING achieves the best results on all the datasets. In particular, when the number of layers takes a large value, the performance may be degraded due to the introduction of too much useless information.

Table 4: Impact of the number of layers H, where the best results are marked in bold.

Dataset	Amazon-book		Yelp2018		Last-FM	
Metrics	R@20	N@20	R@20	N@20	R@20	N@20
PING-1	0.2937	0.1621	0.1905	0.1136	0.1636	0.0866
PING-2	**0.3043**	**0.1759**	**0.2132**	**0.1336**	**0.1752**	**0.1014**
PING-3	0.2032	0.1010	0.1038	0.0435	0.1267	0.0646

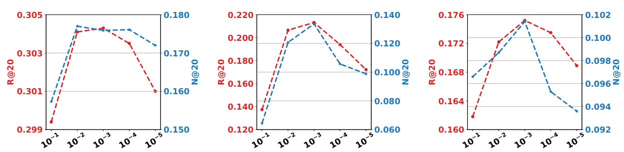
5.6.2 Impact of the Regularization Weight. We next consider the effect of the weight of the parameter regularization term, where its values are set to $1e^{-1}$, $1e^{-2}$, $1e^{-3}$, $1e^{-4}$, and $1e^{-5}$, respectively. The result is shown in Fig. 7. We can observe that when the regularization weight takes a moderate value, our PING can have a better result on all the datasets. On the contrary, if its value is set to larger or smaller, it will cause damage to the performance of our PING.

5.6.3 Impact of the Number of Shared Intent. Since relational intent is a key component in the proposed user intent fusion, which carries the semantic information derived from the knowledge graph to provide a finer-grained user intent, the number of shared intents it



(a) Amazon-book (b) Yelp2018 (c) Last-FM

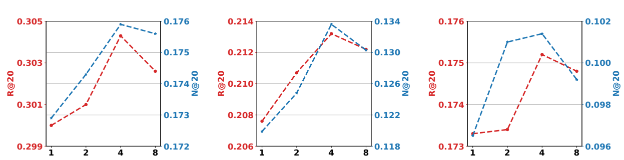
Figure 6: Recommendation performance of GCM, UEG, and our PING with different downstream models, i.e., MF, FM, and MLP, on the three datasets.



(a) Amazon-book (b) Yelp2018 (c) Last-FM

Figure 7: Impact of the value of the regularization weight. Best viewed in color.

contains clearly also has an impact on the performance of our PING. We consider the cases where the number of shared intent is set to 1, 2, 4, and 8 respectively, and report the corresponding results in Fig. 8. As shown in Fig. 8, when the value of $|\mathcal{P}|$ is set to 1, 2, and 4, we can find that the performance of our PING gradually improves as the number of shared intents increases. However, when $|\mathcal{P}|$ is set to a larger value, our PING performance will drop significantly. Therefore, it is important to choose a reasonable number of shared intents in practice.



(a) Amazon-book (b) Yelp2018 (c) Last-FM

Figure 8: Impact of the number of shared intent ($|\mathcal{P}|$). Best viewed in color.

6 CONCLUSIONS

In this paper, aiming at the underutilization of knowledge graphs for the context-aware recommendation, we propose a pairwise intent graph embedding learning (PING) framework to effectively utilize knowledge graphs to enhance the high-order information collaboration of users, items, and contexts. Our PING includes three modules, i.e., a graph construction module for obtaining the pairwise intent graph, a pairwise intent joint graph convolution module for refining the embeddings of all features, and a recommendation module for applying the refined embeddings to improve recommendation performance. Finally, we conduct extensive experiments on three real-world datasets to verify the effectiveness and compatibility of our PING.

ACKNOWLEDGMENTS

We thank the support of the National Natural Science Foundation of China Nos. 61836005, 62272315, and 62172283.

REFERENCES

[1] Gediminas Adomavicius, Konstantin Bauman, Alexander Tuzhilin, and Moshe Unger. 2022. Context-aware recommender systems: From foundations to recent developments. In *Recommender Systems Handbook*. 211–250.

[2] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. 2011. Matrix factorization techniques for context aware recommendation. In *Proceedings of the 5th ACM Conference on Recommender Systems*. 301–304.

[3] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *Proceedings of The Web Conference 2019*. 151–161.

[4] Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient non-sampling factorization machines for optimal context-aware recommendation. In *Proceedings of The Web Conference 2020*. 2400–2410.

[5] Zhang Chuanyan and Hong Xiaoguang. 2021. Neural graph filtering for context-aware recommendation. In *Proceedings of the 13th Asian Conference on Machine Learning*. 969–984.

[6] Xichen Ding, Jie Tang, Tracy Liu, Cheng Xu, Yaping Zhang, Feng Shi, Qixia Jiang, and Dan Shen. 2019. Infer implicit contexts in real-time online-to-offline recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2336–2346.

[7] Yuntao Du, Xinjun Zhu, Lu Chen, Baihua Zheng, and Yunjun Gao. 2022. HAKG: Hierarchy-aware knowledge gated network for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1390–1400.

[8] Tianyi Gu, Kaiwen Huang, Jie Zhang, Kai Zhang, and Ping Li. 2023. Fast convolutional factorization machine with enhanced robustness. *IEEE Transactions on Knowledge and Data Engineering* 35, 3 (2023), 2579–2589.

[9] Wei Guo, Can Zhang, Huifeng Guo, Ruiming Tang, and Xiuqiang He. 2020. Multi-branch convolutional network for context-aware recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1709–1712.

[10] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 355–364.

[11] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: N-dimensional tensor factorization for

context-aware collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*. 79–86.

[12] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[13] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1754–1763.

[14] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2181–2187.

[15] Dugang Liu, Mingkai He, Jinwei Luo, Jiangxu Lin, Meng Wang, Xiaolian Zhang, Weike Pan, and Zhong Ming. 2022. User-event graph embedding learning for context-aware recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1051–1059.

[16] Yahui Liu, Furao Shen, and Jian Zhao. 2019. Pairwise interactive graph attention network for context-aware recommendation. *arXiv preprint arXiv:1911.07429* (2019).

[17] Lei Mei, Pengjie Ren, Zhumin Chen, Liqiang Nie, Jun Ma, and Jian-Yun Nie. 2018. An attentive interaction network for context-aware recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 157–166.

[18] Haithem Mezni, Djamal Benslimane, and Ladjel Bellatreche. 2021. Context-aware service recommendation based on knowledge graph embedding. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5225–5238.

[19] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 635–644.

[20] Yue Shi, Martha Larson, and Alan Hanjalic. 2013. Mining contextual movie similarity with matrix factorization for context-aware recommendation. *ACM Transactions on Intelligent Systems and Technology* 4, 1 (2013), 1–19.

[21] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 297–305.

[22] Moshe Unger and Alexander Tuzhilin. 2022. Hierarchical latent context representation for context-aware recommendations. *IEEE Transactions on Knowledge and Data Engineering* 34, 7 (2022), 3322–3334.

[23] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: Knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 109–118.

[24] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 968–977.

[25] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 950–958.

[26] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of The Web Conference 2021*. 878–887.

[27] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. 5329–5336.

[28] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. CKAN: Collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 219–228.

[29] Jiancan Wu, Xiangnan He, Xiang Wang, Qifan Wang, Weijian Chen, Jianxun Lian, and Xing Xie. 2022. Graph convolution machine for context-aware recommender system. *Frontiers of Computer Science* 16, 6 (2022), 1–12.

[30] Xin Xin, Bo Chen, Xiangnan He, Dong Wang, Yue Ding, and Joemon M Jose. 2019. CFM: Convolutional factorization machines for context-aware recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3926–3932.

[31] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1434–1443.

[32] Yu Yang, Jiangxu Lin, Xiaolian Zhang, and Meng Wang. 2022. PKG: A personal knowledge graph for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3334–3338.

[33] Fajie Yuan, Guibing Guo, Joemon M Jose, Long Chen, Haitao Yu, and Weinan Zhang. 2016. Optimizing factorization machines for top-N context-aware recommendations. In *Proceedings of the 17th International Conference on Web Information Systems Engineering*. 278–293.

[34] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 353–362.

[35] Lan Zhang, Xiang Li, Weihua Li, Huali Zhou, and Quan Bai. 2021. Context-aware recommendation system using graph-based behaviours analysis. *Journal of Systems Science and Systems Engineering* 30 (2021), 482–494.

[36] Yong Zheng, Bamshad Mobasher, and R. Burke. 2013. The role of emotions in context-aware recommendation. In *Workshop on Human Decision Making in Recommender Systems (Decisions@RecSys' 13)*.

[37] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2014. Deviation-based contextual SLIM recommenders. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 271–280.

[38] Jin Peng Zhou, Zhaoyue Cheng, Felipe Pérez, and Maksims Volkovs. 2020. TAFA: Two-headed attention fused autoencoder for context-aware recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 338–347.