Interdependent-path Recurrent Embedding For Knowledge Graph-aware Recommendation

Xiao Sha*, Jianwen Wang, Xiaoran Xu, and Jianchuan Ding

Department of Computer Science, Hebei University of Water Resources and Electric Engineering, Cangzhou, 061001, China

* *Corresponding author. E-mail: E-mail:shaxiao@hbwe.edu.cn*

Received: Mar. 03, 2025; Accepted: May. 20, 2025

Knowledge graphs (KGs) have demonstrated their effectiveness in providing high-quality recommendations by incorporating rich semantic relationships between entities. However, existing KG-aware recommendation methods face significant challenges in sufficiently exploiting both the structural and semantic information while maintaining computational efficiency. We propose the Interdependent-path Recurrent Embedding (IPRE) framework that addresses these limitations through novel interdependent path construction and attentive encoding. The framework automatically generates interdependent paths connecting user-item pairs, preserving both semantic relationships and topological dependencies with linear time complexity. A dedicated attentive recurrent network then encodes these paths by learning relation-aware representations and adaptively weighting different predecessors' influence. Comprehensive experiments on three real-world datasets demonstrate IPRE's superiority, achieving average improvements of 8.79% in Hit ratio and 9.40% in NDCG over state-of-the-art methods. The framework shows particular effectiveness in sparse data scenarios, while maintaining competitive computational efficiency. These results validate IPRE's capability to effectively transform KG information into accurate recommendations through its innovative path modeling approach.

Keywords: Recommender Systems; Knowledge Graphs; Attention Mechanism; Collaborative Filtering

http://dx.doi.org/10.6180/jase.202603_29(3).0004

1. Introduction

Knowledge graphs (KGs) serve as diverse information sources and have been widely utilized to enhance recommendation systems [1]. In this context, the connections between users and items illustrate their fundamental relationships.Recent KG-aware recommendation methods model these connectivities in various ways, such as path-based [2–4], propagation-based [5–7], and subgraph-based methods [8]. However, these methods either fail to sufficiently exploit the KGs for enhanced recommendations or suffer from high computational complexity.

Figure 1 illustrates a sample knowledge graph within the context of movies. To infer Jane's preference towards Titanic, path-based methods first extract several paths linking Jane and Titanic from the KG (e.g., p_1 - p_4), and then independently model their impacts on Jane's taste for Titanic. Nevertheless, these separate paths (i.e., linear sequences) fail to capture the rich interdependencies among paths, which are the links originally existing in the complex graph [9]. For instance, the links 'Jane → Ben' and 'Superman → Ben' are ignored in p_1, while they represent rich semantics and topology of the connectivity between Jane and Titanic. Due to the ignorance of such path interdependencies, path-based methods fail to provide a complete description of the user-item connectivities. In contrast, propagation-based methods model the connectivity between Jane and Titanic by iteratively propagating information over the entire KG. The preference of Jane on Titanic is learned by aggregating information from all of her

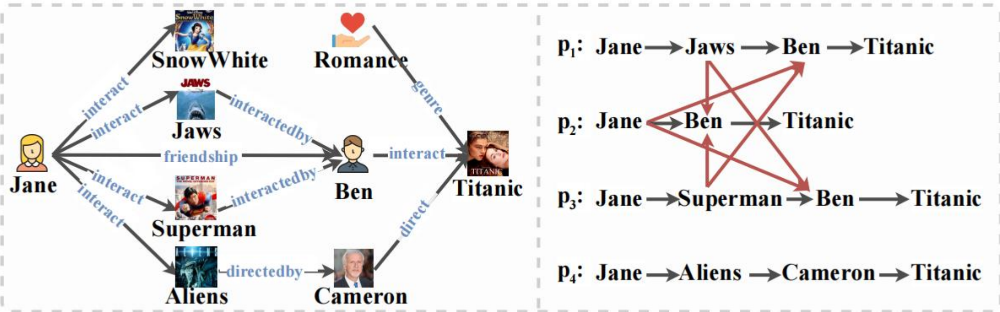

Fig. 1. A representation of a KG in the film sector, demonstrating entities like users and movies, as well as the relations between these entities, such as interactions and friendships.

neighbors (e.g., Snow White, Jaws, and Aliens). Nevertheless, such information may contain noise (e.g., Snow White, Aliens, and Romance) that is irrelevant to this specific interaction, which misleads the inference of Jane's interest in Titanic.

To ease these issues, subgraph-based methods assemble the qualified paths connecting Jane and Titanic into a subgraph, which is then encoded for estimating Jane's preference. By considering the path interdependencies, the subgraph is more expressive for characterizing the connectivity between Jane and Titanic. Nevertheless, constructing and encoding such subgraphs for all user-item pairs is computationally expensive in large-scale KGs. In particular, it might be resource-wasting to employ subgraphs where the dependencies among paths that connect user-item pairs are extremely weak (i.e., independent paths). Due to the high computational complexity, subgraph-based methods fail to explore large-scale KGs efficiently.

Considering the limitations of existing methods, we believe it is vital to sufficiently exploit KGs in an efficient manner. Towards this end, we introduce an interdependent-paths (I-paths) structure which benefits recommendation in two aspects: (1) the I-paths capture both the semantics and topology of KGs by considering the rich interdependencies among paths that connect user-item pairs; and (2) the I-paths are more efficient than subgraphs, as they can be constructed and encoded with less computational cost. However, it is non-trivial to employ I-paths due to two major challenges: (1) how to construct the I-paths connecting user-item pairs in a large-scale KG; and (2) how to effectively encode the I-paths for enhanced user preference inference.

In this paper, we devise an *Interdependent-paths Recurrent Embedding* (IPRE) framework, equipped with two designs to address these challenges: (1) I-paths construction, which generates the I-paths connecting user-item pairs by incorporating the interdependencies among paths; and (2) an attentive recurrent network, which encodes the I-paths via gated recurrent unit networks (GRUs) with an attention mechanism. Specifically, GRUs are employed to learn enhanced entity embeddings by capturing the influence between entities and the topology of I-paths, while the attention mechanism discriminates the importance of entity relations to retain salient semantics. To the best of our knowledge, this is the first time that I-paths have been proposed to characterize user-item connectivities for KG-aware recommendation. Extensive experiments demonstrate the superiority of IPRE over state-of-the-art methods and its effectiveness in mitigating the data sparsity issue.

2. Methods

In this section, we first review existing KG-aware recommendation methods. We then formally define the recommendation task and present the proposed IPRE framework. Finally, we describe the core modules of IPRE and their functionalities in detail.

2.1. Related works

Recent advances in KG-aware recommendation systems can be broadly categorized based on their strategies for modeling user-item connectivities, each exhibiting distinct strengths and limitations.

Direct-relation based methods [10–14] learn embeddings from immediately connected entities but generally fail to capture multi-hop semantic information. The recent CKNNS [15] addresses this limitation by effectively capturing high-order collaborative signals among items through a gated aggregation mechanism. However, its dependence on neighborhood information may restrict its effectiveness in scenarios characterized by data sparsity or cold-start users.

Path-based methods [4, 16–18] explicitly model connectivity paths, evolving from manual meta-path design to automated path exploration. Early works such as KPRN [2] and RKGE [3] utilized recurrent neural networks (RNNs) to encode independent paths, neglecting critical inter-path dependencies. More recent approaches like KGIN [19] incorporate path interaction modeling, while CPER [20] integrates counterfactual reasoning with reinforcement learning to enhance path evaluation robustness. Despite these advances, effectively modeling cross-path dependencies remains a significant challenge.

Propagation-based methods [21, 22] iteratively perform propagation over the entire KG to assist in recommendatio. KGAT [23] employs graph attention mechanisms, and KGCL [24] applies contrastive learning to mitigate noise. LightGODE [25] performs graph convolution post-training to enhance efficiency, achieving substantial speedups without compromising accuracy. Nevertheless, noise propagation from distant neighbors in dense KGs continues to adversely affect performance.

Subgraph-based methods explore local structures with enhanced expressiveness. KGCL [24] employs contrastive learning on augmented knowledge subgraphs to suppress noise and distill robust item semantics, but its reliance on global graph traversals introduces computational overhead. More recent work HKSAR [8] encodes hierarchical subgraphs but suffers from expensive graph traversal, limiting scalability.

These limitations motivate our proposed IPRE framework, which addresses three critical gaps: (1) modeling cross-path dependencies through hierarchical path encoding; (2) adaptive noise reduction via relation-aware attention mechanisms; and (3) achieving linear time complexity through optimized path sampling.

2.2. Task formulation and overview of IPRE

Notations. We denote the user and item sets as $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$ and $\mathcal{I} = \{i_1, i_2, \cdots, i_m\}$, respectively. Each entry $r_{u,i}$ in the user-item feedback matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ is defined as: $r_{u,i} = 1$ if the interaction between user u and item i is observed, and 0 otherwise. For generality, we use "entity" to refer to objects (e.g., user, movie, and genre) that can be mapped into a KG. The KG and the investigated task in this paper are described below.

Knowledge Graph. Let $\mathcal{E}, \mathcal{L}$ denote the sets of entities and links, respectively. A KG is defined as a directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{L})$ composed of entity-relation-entity triples $\{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{L}\}$; each triple describes a relation r from the head entity h to the tail entity t. For convenience, h is also denoted as the *predecessor* of t.

KG-aware Top-N Recommendation. Given the KG $\mathcal{G}$, for each user $u \in \mathcal{U}$, our task is to generate a ranked list of N items that will be of interest to user u.

Figure 2 presents the overall framework of IPRE, which is composed of three modules: (1) I-paths Construction – it constructs the I-paths that connect user-item pairs to represent their connectivities; (2) I-paths Encoding – the I-paths are further encoded via an attentive recurrent network to learn embeddings for users, items, and I-paths; and (3) Multi-layer Perceptron (MLP) Prediction – by feeding the embeddings of users, items, and I-paths as input, it uses nonlinear layers to predict user preferences towards items.

2.3. I-paths construction

Given the KG $\mathcal{G}$, we sample the paths connecting user-item pairs and then construct the I-paths by adding the interdependencies among the sampled paths.

Path Sampling. It is not feasible to extract all the paths between user-item pairs from KGs, since the number of paths grows exponentially with the increasing path length [2]. Paths with a short length (no more than six) are sufficient to model user-item connectivities, whereas long paths may bring in remote neighbors with noise [3]. For efficiency, we uniformly sample K paths connecting a user-item pair with a length of up to six and denote the path set as $\mathcal{P}_{u,i}$. Note that we employ uniform path sampling for simplicity and leave the exploration of non-uniform samplers (e.g., importance sampling [5]) as future work.

Interdependencies Injection. By injecting the interdependencies among paths in $\mathcal{P}_{u,i}$, we construct the I-paths $\mathcal{T}_{u,i}$ for (u, i). In particular, we first traverse $\mathcal{P}_{u,i}$ and identify all the entities that have multiple predecessors. Hereafter, for each entity with multiple predecessors in a path, we inject directed links from its predecessors in other paths as the interdependencies. Take the path p_1 in Figure 2(b) as an example, we add links from Jane (in p_2) and Superman (in p_3) to Ben as the interdependencies. After performing such an operation for each path, we obtain the I-paths structure $\mathcal{T}_{u,i}$ and further represent it by a topological sorting to preserve the structural information, which is a linear ordering of its entities such that for each directed link from entity e_k to e_l, e_k comes before e_l in the ordering [26]. For instance, the topological sorting of the I-paths between Jane and Titanic is shown in Figure 2(b). This ordering is then utilized

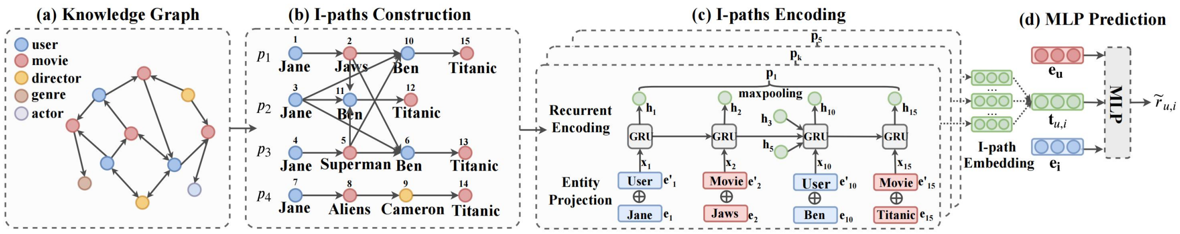

Fig. 2. The overall framework of IPRE describing the case of a user-item pair, which consists of three modules: I-paths Construction, I-paths Encoding, and MLP Prediction. Figure 2(c) presents three layers of Attentive Recurrent Network: Entity Projection Layer, Recurrent Encoding Layer, and I-path Embedding Layer.

to guide the embedding learning process in the attentive recurrent network, as introduced next.

2.4. I-paths encoding

The constructed I-paths are encoded via the attentive recurrent network, which is able to fully capture the interdependencies among paths. This is achieved via three layers, namely entity projection layer, recurrent encoding layer with an attention mechanism, and I-paths encoding layer.

Entity Projection Layer. Given the I-paths $\mathcal{T}_{u,i}$ as input, we first project the entity e_l (e.g., Jane) and entity type e'_l (e.g., user) into two low-dimensional vectors $\mathbf{e}_l \in \mathbb{R}^d$ and $\mathbf{e}'_l \in \mathbb{R}^{d'}$, where d and d' are the respective embedding sizes. Afterwards, we concatenate the entity embedding ($\mathbf{e}_l$) and the entity type embedding ($\mathbf{e}'_l$) as the feature vector of e_l, given by:

$$\mathbf{x}_l = \mathbf{e}'_l \oplus \mathbf{e}_l, \tag{1}$$

where $\oplus$ is the concatenation operation; and $\mathbf{x}_l$ is the feature vector of e_l, which is then fed into the recurrent encoding layer as the input state of e_l to learn its embedding. In this way, the heterogeneity of entity types is incorporated into the learned entity embeddings, encoding more information from KGs. Moreover, as parameter initialization is vital for deep learning models [27], we initialize $\mathbf{e}_l$ with pre-trained embeddings via TransR [28].

Recurrent Encoding Layer. To effectively encode $\mathcal{T}_{u,i}$, we take inspiration from the gated recurrent unit networks (GRUs) which have superior capability on capturing semantic dependencies between words [9]. Specifically, given the input state $\mathbf{x}_l$, GRUs update the hidden state $\mathbf{h}_l$ to help capture the influence between entities along $\mathcal{T}_{u,i}$. We additionally fuse the heterogeneous relations of KGs since an entity may connect with multiple predecessors via various relations (e.g., friendship, interaction), indicating different semantics. Such rich semantics encoded in relations are critical for understanding the diverse user intents [2].

In particular, for each predecessor entity e_k, $\mathbf{h}_k$ denotes its hidden state vector at the previous time step in the GRU computation. We represent the relation $r_{k,l}$ from the predecessor e_k to the target entity e_l as the embedding $\mathbf{r}_{k,l} \in \mathbb{R}^{d''}$, where d'' is the embedding size. The relation embedding $\mathbf{r}_{k,l}$ is used to enhance the hidden state of the predecessor e_k (i.e., $\mathbf{h}_k$) as follows:

$$\hat{\mathbf{h}}_k = f(\mathbf{h}_k \oplus \mathbf{r}_{k,l}) \qquad \forall e_k \in \mathcal{N}_l, \tag{2}$$

where $f = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$; $\mathcal{N}_l$ denotes the set of predecessors of e_l; and the output $\hat{\mathbf{h}}_k$ is the relation-enhanced hidden state of e_k, which condenses the information from both the predecessor e_k and the relation $r_{k,l}$. Initially, for entities without predecessors (i.e., leaf nodes in the path), $\mathbf{h}_k$ is initialized as a zero vector, following the standard practice in RNNs and their variants such as GRUs and LSTMs [29]. For other entities, $\mathbf{h}_k$ is computed recursively via GRU updates along the path.

The relation-enhanced hidden states of predecessors are then aggregated as follows:

$$\mathbf{h}_{\mathcal{N}_l} = \textstyle\sum_{e_k \in \mathcal{N}_l} \alpha_{k,l} \cdot \hat{\mathbf{h}}_k \quad \forall e_k \in \mathcal{N}_l, \tag{3}$$

where $\alpha_{k,l}$ is the attention weight of e_k, which specifies the influence of e_k on e_l and will be introduced in detail later; $\mathbf{h}_{\mathcal{N}_l}$ is thus the attentive combination of predecessors' hidden states for e_l, and utilized as the previous hidden state of e_l to update its embedding via the GRUs:

$$\mathbf{h}_l = \mathrm{GRU}\left(\mathbf{h}_{\mathcal{N}_l}, \mathbf{x}_l\right), \tag{4}$$

where $\mathbf{h}_l$, $\mathbf{h}_{\mathcal{N}_l}$ are the current hidden state and previous hidden state of e_l, respectively; and $\mathbf{x}_l$ is the input state of e_l. Once obtaining the embedding (i.e., current hidden state) for each entity in $\mathcal{T}_{u,i}$, we compute the embedding for each path $p_j \in \mathcal{T}_{u,i}$ by:

$$\mathbf{p}_j = f_g(\{\mathbf{h}_l : e_l \in p_j\}), \tag{5}$$

In Equation (5), we aggregate the embeddings of entities in path $\mathbf{p}_j$ to obtain the path embedding $\mathbf{p}_j$. Subsequently, the path embeddings $\mathbf{p}_j$ are integrated to form a unified representation for $\mathcal{T}_{u,i}$ through the I-paths encoding layer.

Attention Mechanism. Recognizing that various predecessors can exert differing influences on the target entity e_l, we employ an attention mechanism to effectively differentiate their significance, expressed as follows:

$$\widetilde{\alpha}_{k,l} = \mathbf{q}^{\mathbf{T}} \cdot \tanh\left(\mathbf{W} \cdot \hat{\mathbf{h}}_k + \mathbf{b}\right), \tag{6}$$

where $\widetilde{\alpha}_{k,l}$ is the attention score for predecessor e_k; $\mathbf{q}$ is the attention vector; tanh is the activation function; $\mathbf{W}$ is the transformation matrix and $\mathbf{b}$ is the bias vector. In particular, we first transform the relation-enhanced hidden state $\hat{\mathbf{h}}_k$ by a nonlinear layer, and then measure $\widetilde{\alpha}_{k,l}$ as the similarity between the transformed $\hat{\mathbf{h}}_k$ and $\mathbf{q}$. We further normalize $\widetilde{\alpha}_{k,l}$ across all predecessors of e_l via the softmax function:

$$\alpha_{k,l} = \frac{exp(\widetilde{\alpha}_{k,l})}{\sum_{e_j \in \mathcal{N}_l} exp(\widetilde{\alpha}_{j,l})} \tag{7}$$

where $\alpha_{k,l}$ is the final attention weight of e_k. By assigning the informative predecessor a larger weight, the attention mechanism effectively captures the prominent relations, facilitating the learning of improved embeddings.

I-paths Encoding Layer. The $\mathcal{T}_{u,i}$ connecting the user u and item i encapsulates the fundamental reasoning behind their interaction. Nevertheless, previous research often overlooks this essential information, focusing solely on user and item embeddings to predict user preferences [30].

To fully exploit $\mathcal{T}_{u,i}$ for enhanced recommendation, we thus generate a unified embedding for $\mathcal{T}_{u,i}$ by performing a weighted pooling operation over all path embeddings ($\mathbf{p_j}$):

$$\mathbf{t}_{u,i} = \text{max-pooling}(\{\mathbf{p}_j \cdot e^{-\beta \times |p_j|}\} : p_j \in \mathcal{T}_{u,i}) \tag{8}$$

where $|p_j|$ is the length of path p_j; The hyperparameter $\beta > 0$ serves to balance the trade-off associated with path length; and $\mathbf{t}_{u,i}$ is the unified embedding of $\mathcal{T}_{u,i}$, which condenses the holistic semantics and topology of $\mathcal{T}_{u,i}$.

2.5. MLP prediction

By employing the effectively trained embeddings for the user ($\mathbf{e}_u$), the item ($\mathbf{e}_i$), and the I-paths that connect them ($\mathbf{t}_{u,i}$), we predict the user preference as follows:

$$\widetilde{r}_{u,i} = \text{MLP}\left(\mathbf{e}_u \oplus \mathbf{t}_{u,i} \oplus \mathbf{e}_i\right) \tag{9}$$

where $\widetilde{r}_{u,i}$ is the predicted score for user u regarding item i. In Equation (9), we input the concatenated embeddings of u, $\mathcal{T}_{u,i}$, and i into a multi-layer perceptron (MLP) to capture the complex interactions between users and items. This approach fully exploits the rich semantics and underlying topology of I-paths for improved recommendation performance.

2.6. Model training and optimization

Objective Function. Consistent with the approach in [27], we frame the top-N recommendation problem as a binary classification task, where a target value of 1 signifies an observed user-item interaction, and 0 indicates its absence. Specifically, we utilize the negative log-likelihood as our objective function:

$$\mathcal{J} = -\sum_{(u,i)\in\mathcal{R}^+} \log \widetilde{r}_{u,i} - \sum_{(u,j)\in\mathcal{R}^-} \log(1 - \widetilde{r}_{u,j}) \tag{10}$$

where $\mathcal{R}^+$ and $\mathcal{R}^-$ denote the sets of observed and unobserved user-item interactions, respectively. For each user, we uniformly sample negative items that have not been interacted with, maintaining a sampling ratio of 4:1 in relation to the number of positive items. The parameters are optimized using backpropagation through time (BPTT) [29] within the attentive recurrent network, while standard backpropagation is employed in other components.

Complexity Analysis. The primary time expenditure of IPRE is attributed to two factors: (a) the construction of I-paths and (b) the encoding of I-paths using the attentive recurrent network. For (a), it takes $O(K^2P)$, where K and P are the total number and average length of the sampled paths, respectively. For (b), the computational complexity for a user-item pair is $O(KP\overline{p}(d + d')d)$, where $\overline{p}$ is the average number of predecessors of an entity in $\mathcal{T}_{u,i}$; d and d' are the embedding sizes of the entity and entity type, respectively. The total training complexity is given by $O(|\mathbf{R}|K^2P + |\mathbf{R}|KP\overline{p}d^2)$, where $|\mathbf{R}|$ is the number of observed user-item interactions. In practice, $K, P, \overline{p} \ll |\mathbf{R}|$. Hence, the complexity of IPRE is linear with respect to $|\mathbf{R}|$ and quadratic with respect to the embedding size d.

Practical Considerations on K and P. Although the number of sampled paths K and the path length P are generally much smaller than $|\mathbf{R}|$, it is important to note that overly large values of K or P can substantially increase computational costs due to the $O(K^2P)$ term in path construction. For instance, in sparse graphs such as social networks with limited user interactions, a larger K may be necessary to obtain a sufficient number of informative paths, which can lead to increased runtime. Similarly, in hierarchical or sequential data (e.g., dependency parsing in NLP), paths may span multiple levels, resulting in longer P values that not only amplify computational demands but may also introduce noise.

To ensure computational feasibility, we recommend the following guidelines: (a) typically, setting K within the range of 10 to 50 achieves a reasonable balance between efficiency and performance in most scenarios. In cases of extremely sparse graphs, K can be increased cautiously, while carefully monitoring the quadratic increase in computational complexity; (b) restricting P to a moderate length (e.g., $P \leq 6$) helps mitigate noise from overly long paths and reduces computational overhead [2]. Longer paths should be considered only when supported by domain knowledge indicating their relevance.

3. Results and discussion

We perform extensive experiments on three real-world datasets to address the following research questions:
RQ1: Does IPRE demonstrate superior performance compared to state-of-the-art methods regarding recommendation accuracy and time complexity? Additionally, can it provide enhanced recommendation results across varying levels of data sparsity?
RQ2: In what ways do the critical design elements of IPRE influence its recommendation performance?
RQ3: How do different hyperparameter configurations impact the performance of IPRE?

3.1. Experimental setup

Datasets. We utilize three datasets: (1) MovieLens-1M[1], a well-known dataset for movie recommendations that includes user ratings on a scale from 1 to 5; (2) Last-FM[2], a music listening dataset sourced from the Last.fm online music platform, where tracks are treated as items; and (3) Yelp[3], which contains user ratings for businesses, also on a scale from 1 to 5. This dataset further includes social relationships and business attributes such as category and city.

The datasets are processed in accordance with the methodology outlined in [31]: if a user rates an item, it is recorded as an observed interaction with a value of 1; otherwise, it is assigned a value of 0. Besides user-item interactions, we merge more information into KGs for each dataset. We combine MovieLens-1M with IMDb[4] as MI-1M by linking the titles and release dates of movies, to obtain side information about movies, such as genres, actors, and directors. For Last-FM, we map tracks into objects in the database called Freebase via title matching to get attributes of tracks, such as artists, engineers, producers, versions, types, contained_songs, etc. For Yelp, we extract knowledge from the social network and business information network (e.g., category, city).

Evaluation Protocols. We employ the *leave-one-out* strategy, which has been extensively utilized in prior studies [2, 27], to assess recommendation performance. For each user, the most recent interaction is set aside as the test set, while the remaining interactions are used to form the training set. Consistent with the approaches in [2, 27, 30], during the testing phase, we randomly select 100 items that the user has not previously interacted with, and then rank the test item among these 101 items to streamline the testing process. In line with [2, 27], we utilize Hit@$N$and NDCG@N as our evaluation metrics, calculating both for each test user and reporting the average scores. Generally, higher values for these metrics signify improved ranking accuracy.

Table 1. Statistics of the datasets.

		MI-1M	Last-FM	Yelp
User-Item Interactions	#Users	6,040	23,566	37,940
	#Items	3,382	48,123	11,516
	#Interactions	756,684	3,034,796	229,178
	Data Density	3.704%	0.268%	0.052%
Knowledge Graph	#Entities	18,920	138,362	46,606
	#Relation Types	10	10	6
	#Links	968,038	2,007,423	302,937
	Graph Density	0.270%	0.011%	0.014%

Comparison Methods. We compare IPRE with five types of baseline methods.

- Plain Collaborative Filtering (CF) recommenders: (a) **BPRMF** [32] - a classic algorithm that minimizes a matrix factorization (MF) model using pairwise ranking loss; (b) **NeuMF** [27] - a state-of-the-art deep learning based method that combines MF with an MLP model; (c) **LightGCN++** [33] - a GNN-based model that incorporates flexible embedding norm scaling and adaptive neighbor weighting to enhance recommendation accuracy.

- Direct-relation based models: (d) **CKE** [34] - a notable method that enhances MF by utilizing embeddings generated from the TransR model [28].

- Path-based models: (e) **MCRec** [30] - it extracts meta-paths between user-item pairs to represent their connectivities and encodes these paths using convolutional neural networks; (f) **KPRN** [2] - it incorporates entity types into the learning of path embeddings and performs weighted pooling across these embeddings to predict user preferences; (g) **S4Rec** [18] - it fuses semantic meta-paths and structural information via a hybrid graph-SVD framework.

- Propagation-based models: (h) **KGAT** [23] - it utilizes GNNs to disseminate information across the entire KG, generating user and item embeddings, and employs their inner products to approximate user preferences; (i) **DiffKG** [35] - it leverages generative diffusion and

[1]https://grouplens.org/datasets/movielens/
[2]https://grouplens.org/datasets/hetrec-2011/
[3]http://www.yelp.com/dataset-challenge
[4]https://www.imdb.com/.

collaborative signals to refine item embeddings for accurate recommendation.

- Subgraph-based method with KGs:(j) **KGCL** [24] - it utilizes augmented knowledge subgraphs to suppress noise and extract robust item semantics; (k) **HKSAR** [8] - it extracts subgraphs that connect user-item pairs to characterize their connectivities, which are then encoded using GNNs to predict user preferences.

Hyper-parameter Settings. The optimal hyper-parameter settings for all comparison methods are achieved by either empirical study or following suggestions in original papers. For the IPRE model, we utilize the Adam optimizer [36] and conduct a grid search over the set $\{0.001, 0.002, 0.01, 0.02\}$ to determine the optimal learning rate γ. The best value for the L_2 regularization coefficient λ is explored within the range $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. The optimal configurations for the remaining hyperparameters are as follows: a batch size of 256; embedding dimensions for entity, entity type, and relation set to $d = 128$, $d' = d'' = 32$; a hidden state size of 128; a multi-layer perceptron (MLP) with 3 layers; the number of sampled paths K for each user-item pair is set to 20, 30, and 30 for the MI-1M, Last-FM, and Yelp datasets, respectively; and a trade-off parameter β fixed at 0.1 across all three datasets.

3.2. Performance comparison (RQ1)

Overall Performance Comparison. Table 2 presents the performance comparison results on three datasets for $N = 5$ and $N = 10$. The key observations are summarized as follows.

First, most KG-aware methods outperform traditional CF-based approaches, demonstrating the advantages of integrating KGs to enhance recommendation quality. Among CF-based methods, NeuMF surpasses the classic BPRMF by leveraging deep neural networks to capture complex non-linear user-item interactions more effectively. Moreover, LightGCN++, a recent advancement in graph-based recommendation, further improves upon NeuMF, underscoring the benefits of optimized graph convolutional networks in boosting recommendation accuracy.

Within path-based methods, KPRN and S4Rec show substantial improvements over earlier approaches such as MCRec, which heavily depend on the quality of meta-paths to model user-item relationships. This indicates that more sophisticated path encoding and fusion techniques can better exploit the semantic structure of KGs.

The propagation-based model DiffKG achieves competitive results by refining item embeddings via generative diffusion processes combined with collaborative signals. However, it is generally outperformed by the subgraph-based method HKSAR, which extracts and encodes user-item connecting subgraphs to obtain more robust representations of connectivity. Despite HKSAR's strong performance, it suffers from relatively high computational costs (as shown in Table 3).

In summary, the proposed IPRE consistently outperforms all baseline methods across the three datasets. Notably, it achieves an average improvement of 8.79% and 9.40% over the second-best baseline in terms of Hit@ N and NDCG@N, respectively. These enhancements are statistically significant, as validated by a standard paired t-test conducted over 20 rounds of results [37], yielding a p-value of less than 0.05. This evidence supports the effectiveness of IPRE in delivering personalized recommendations by leveraging the intricate I-paths within KGs, thereby enhancing the user experience in recommendation systems.

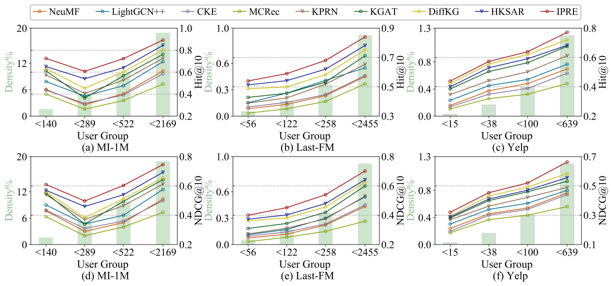

Fig. 3. Performance comparison based on varying levels of data sparsity. The histograms in the background represent the interaction density for each user group, while the lines illustrate the performance metrics for Hit@10 and NDCG@10.

Performance Comparison w.r.t. Data Sparsity Levels. We investigate whether leveraging I-paths in KGs can mitigate data sparsity, a key challenge limiting recommender systems' effectiveness. To this end, we conduct experiments on user groups defined by varying levels of interaction sparsity. Following established methods [23, 37], test users are divided into four groups based on their number of training interactions, while keeping the total number of interactions constant across groups. The density of each group is defined as:#total interactions/(# users $\times$ # items). The resulting densities are 1.5%, 5.1%, 9.6%, and 18.9% for MI-1M; 0.05%, 0.17%, 0.35%, and 0.91% for Last-FM; and 0.02%, 0.16%, 0.43%, and 1.19% for Yelp, indicating progressively decreasing sparsity from the first to the last group.

Figure 3 shows Hit@10 and NDCG@10 results across these groups on all datasets, with consistent trends ob-

Table 2. Performance comparison results.'Improve' refers to the advancements made by the IPRE model in comparison to the second-best baseline, indicated by '*'.

Methods	MI-1M				Last-FM				Yelp			
	Hit@N		NDCG@N		Hit@N		NDCG@N		Hit@N		NDCG@N	
	N=5	N=10	N=5	N=10	N=5	N=10	N=5	N=10	N=5	N=10	N=5	N=10
BPRMF	0.421	0.538	0.317	0.383	0.327	0.431	0.251	0.323	0.278	0.369	0.239	0.298
NeuMF	0.582	0.681	0.407	0.483	0.427	0.562	0.347	0.425	0.368	0.481	0.317	0.407
LightGCN++	0.642	0.752	0.498	0.582	0.523	0.658	0.452	0.528	0.487	0.602	0.428	0.512
CKE	0.568	0.659	0.392	0.487	0.438	0.567	0.378	0.431	0.386	0.512	0.329	0.418
MCRec	0.492	0.617	0.391	0.432	0.434	0.568	0.352	0.428	0.369	0.482	0.317	0.403
KPRN	0.667	0.737	0.545	0.613	0.530	0.673	0.452	0.536	0.517	0.640	0.435	0.516
S4Rec	0.603	0.712	0.487	0.562	0.518	0.673	0.463	0.538	0.502	0.628	0.442	0.521
KGAT	0.696	0.761	0.574	0.643	0.581	0.712	0.538	0.608	0.532	0.656	0.484	0.550
DiffKG	0.735	0.792	0.638	0.682	0.662	0.792	0.596	0.639	0.623*	0.735*	0.531*	0.576*
KGCL	0.712	0.776	0.602	0.642	0.628	0.745	0.587	0.631	0.598	0.713	0.511	0.530
HKSAR	0.751*	0.844*	0.638*	0.693*	0.674*	0.798*	0.601*	0.642*	0.612	0.723	0.522	0.556
IPRE	**0.817**	**0.892**	**0.698**	**0.746**	**0.734**	**0.859**	**0.668**	**0.702**	**0.687**	**0.782**	**0.594**	**0.635**
Improve	8.79%	5.69%	9.40%	7.65%	8.90%	7.64%	11.15%	9.35%	10.27%	6.39%	11.86%	10.24%

Table 3. Training Time Comparisons relative to representative baselines.

	MI-1M		Last-FM		Yelp	
	Average	Total	Average	Total	Average	Total
BPRMF	59.4s	1.4h	78.6s	1.9h	10.7s	0.2h
NeuMF	65.2s	1.7h	197.4s	5.2h	13.2s	0.4h
LightGCN++	72.8s	1.9h	156.3s	4.1h	15.9s	0.5h
CKE	68.2s	1.8h	132.4s	3.7h	14.8s	0.5h
MCRec	89.5s	2.3h	178.2s	4.8h	18.3s	0.6h
KPRN	1.7h	20.2h	4.2h	46.8h	0.9h	8.3h
S4Rec	1.2h	14.5h	3.1h	34.2h	0.7h	6.8h
KGAT	108.2s	2.9h	242.6s	6.7h	22.7s	0.7h
DiffKG	224.7s	5.8h	321.5s	8.9h	45.3s	1.3h
KGCL	198.4s	5.1h	298.7s	8.2h	38.2s	1.0h
HKSAR	245.6s	6.7h	364.1s	9.8h	62.5s	1.7h
IPRE	237.5s	6.5h	345.2s	9.6h	50.9s	1.42h

served for other N values. LightGCN++ performs well in denser groups by effectively scaling norms, but its performance deteriorates sharply in sparser groups due to limited capacity to model high-order dependencies without KG auxiliary information. DiffKG excels in moderate-density groups (e.g., 9.6% in MI-1M) but suffers in sparse settings because diffusion amplifies noise. HKSAR captures hierarchical subgraph structures effectively in moderately dense groups, outperforming propagation-based methods; however, it struggles in very sparse groups due to dependence on subgraph connectivity, resulting in performance drops. KGCL achieves competitive results in dense groups (e.g., 18.9% in MI-1M) but underperforms in sparse scenarios, as contrastive learning requires ample positive samples.

Our proposed IPRE consistently surpasses all baselines across all sparsity levels, with average improvements of 9.2% on Hit@10 and 10.1% on NDCG@10 over the nearest competitor. Notably, gains increase as sparsity rises, peaking in the sparsest groups (e.g., 0.02% density in Yelp), where IPRE outperforms DiffKG by 12.3% on Hit@10 and 13.8% on NDCG@10. These results demonstrate IPRE's robustness to sparse data and highlight the advantage of exploiting I-paths to enrich user-item embeddings with complex semantics and higher-order topological information from KGs, particularly benefiting inactive users.

Performance Comparison Regarding Running Time. We evaluate the efficiency of IPRE by analyzing its training duration in comparison to several state-of-the-art recommendation systems. All experiments were conducted on a single NVIDIA GeForce GTX 1080 Ti GPU. The results are presented in Table 3. Path-based methods, such as KPRN and S4Rec, incur significantly higher training costs due to their reliance on exhaustive meta-path enumeration or diffusion processes. In contrast, IPRE demonstrates a complexity level comparable to that of GNN-based recommenders, such as DiffKG and HKSAR, while exhibiting greater efficiency than the path-based approaches. Overall, IPRE offers a favorable trade-off between performance and computational cost, particularly in scenarios that require complex path-based reasoning.

3.3. Model components analysis (RQ2)

In this subsection, we conduct an ablation analysis to investigate the efficacy of the key designs of IPRE, including (1) entity type and relation; (2) attention mechanism; (3) aggregation function f_p; and (4) I-paths embedding.

Impact of Entity Type and Relation. IPRE utilizes both entity types and relations to enhance entity embeddings. To assess their individual impacts, we examine two variants: IPRE-t, which excludes entity type embeddings in Eq.(1), and IPRE-r, which omits relation embeddings in Eq.(2). As shown in Table 4, both variants underperform compared to IPRE. This indicates that integrating diverse entity types and relations significantly improves recommendation performance by enriching the semantic context for better entity embedding learning.

Impact of Attention Mechanism. IPRE incorporates an attention mechanism to differentiate the importance of predecessors during the entity embedding learning process. To study its impact, contrast IPRE with its variant, IPRE-a, which removes the attention mechanism in Eq. (3) and aggregates entity embeddings using the mean-pooling operation instead. As illustrated in Table 4, the performance of IPRE-a significantly deteriorates across all three datasets, with an average decline of 4.84%. This might because treating all predecessors equally introduces noise and misleads the embedding learning process. This also verifies the effectiveness of the attention mechanism in enhancing user preference inference by specifying the varying importance of different predecessors.

Impact of Aggregation Function. In the IPRE model, the function f_p is utilized to aggregate entity embeddings into a path embedding, as defined in Eq.(5). Within this model, f_p is implemented using a max-pooling operation. To explore the effectiveness of alternative aggregation methods, we introduced two variants: IPRE$_{mean}$ and IPRE$_{att}$, which employ mean-pooling and a vanilla attention mechanism for f_p, respectively. The performance analysis presented in Table 4 demonstrates that IPRE consistently outperforms the variants. This suggests that the max-pooling operation successfully identifies the most critical dimensional features, which aids in the effective learning of path embeddings.

Impact of I-paths Embedding. IPRE incorporates the learned I-path embeddings into user preference predictions as shown in Eq. (9). To evaluate their effect, we compare IPRE with the variant IPRE-ip, which excludes I-path embeddings and relies only on user and item embeddings for predictions. As indicated in Table 4, IPRE-ip consistently shows inferior performance across all metrics in the three datasets. This demonstrates that including I-path embeddings in user preference predictions significantly improves recommendation performance by offering essential complementary information to user-item interactions.

3.4. Parameter sensitivity (RQ3)

We finally study how different settings of hyper-parameters affect the performance of IPRE.

Number of Sampled Paths. To investigate the impact of varying the number of sampled paths (K) on model performance, we integrated different sets of paths into the IPRE model, with K values of {5, 10, 20, 30, 40, 50, 60}. As shown in Figure 4(a)(d), optimal performance is achieved with $K = 20$ for the MI-1M dataset and $K = 30$ for the Last-FM and Yelp datasets. This enhancement is due to the increased number of paths (i.e., entities and relations), which allows for the encoding of more comprehensive knowledge from the KGs. However, excessive integration of paths may introduce additional noise, which can detract from model accuracy.

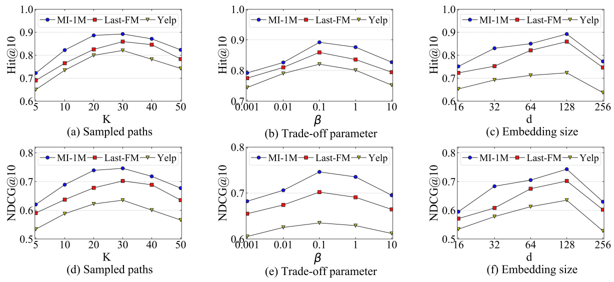

Fig. 4. Hyper-parameter sensitivity on the three datasets.

Trade-off Parameter. Figure 4(b)(e) demonstrates the effects of the trade-off parameter β. Achieving optimal performance with $\beta = 0.1$ across three datasets highlights the critical role of selecting an appropriate β value to effectively penalize the path length.

Embedding Sizes. We assess the impact of embedding size d on IPRE by varying d within the range {16, 32, 64, 128, 256}. Figure 4(c)(f) indicates that the performance of DSR initially increases with the increment in d, peaks at $d = 128$ across the datasets, and subsequently declines as d increases further (e.g., $d = 256$). This suggests that while larger embedding sizes enhance encoding capabilities, they may also negatively affect the model's generalization, potentially leading to overfitting [27].

4. Conclusions

We propose an Interdependent-paths Recurrent Embedding framework to effectively exploit the heterogeneity

Table 4. Ablation study results.'Decrease' refers to the relative performance drop of the respective variants in comparison to IPRE.

	MI-1M				**Last-FM**				**Yelp**			
	Hit	Decrease	NDCG	Decrease	Hit	Decrease	NDCG	Decrease	Hit	Decrease	NDCG	Decrease
IPRE	**0.892**	–	**0.746**	–	**0.859**	–	**0.702**	–	**0.782**	–	**0.635**	–
IPRE-t	0.879	-1.46%	0.734	-1.61%	0.842	-1.98%	0.689	-1.85%	0.771	-1.41%	0.627	-1.26%
IPRE-r	0.873	-2.13%	0.731	-2.01%	0.840	-2.21%	0.685	-2.42%	0.765	-2.18%	0.621	-2.20%
IPRE-a	0.847	-5.04%	0.706	-5.36%	0.822	-4.31%	0.668	-4.84%	0.746	-4.60%	0.604	-4.88%
IPRE-ip	0.824	-7.62%	0.687	-7.91%	0.794	-7.57%	0.651	-6.26%	0.722	-7.67%	0.585	-7.87%
$IPRE_{mean}$	0.858	-3.81%	0.718	-3.75%	0.834	-2.91%	0.679	-3.28%	0.754	-3.58%	0.610	-3.94%
$IPRE_{att}$	0.862	-3.36%	0.723	-3.08%	0.839	-2.33%	0.683	-2.71%	0.761	-2.68%	0.616	-2.99%

of KGs for personalized recommendation. The I-paths construction method autonomously identifies interrelated paths connecting user-item pairs, while the attentive recurrent network enhances embedding learning by jointly capturing the semantic and structural information of KGs. Extensive experimental results demonstrate that IPRE consistently outperforms existing methods, particularly in mitigating the effects of data sparsity.

Despite these strengths, IPRE requires careful hyperparameter tuning, including parameters such as the number of sampled paths K, path length P, and the trade-off parameter β that modulates the influence of path length in the embedding process. Although grid search was employed to find optimal values, performance can vary substantially with different settings, and manual tuning is time-consuming and may not generalize well across diverse datasets. Future research will explore automated machine learning (AutoML) frameworks [38] to automate hyperparameter optimization.

Furthermore, noise and missing information in real-world KGs, such as incorrect triples or incomplete relations, can adversely affect the quality of I-path construction and embedding, thereby reducing recommendation effectiveness. To address this, future work will investigate adversarial training techniques to enhance noise robustness [39]. For example, incorporating adversarial regularization during I-path encoding could simulate noisy triples, thereby enhancing the model's resilience.

Finally, although $\beta = 0.1$ demonstrates strong empirical performance across the datasets studied, its optimal value is likely sensitive to varying KG structural properties such as density, relational complexity, and hierarchy. To address this limitation, future work will explore the application of causal intervention [40] to disentangle the direct and indirect effects of short and long paths on recommendation outcomes. This approach will enable tuning of β based on causal contributions rather than mere correlations, thereby enhancing robustness against noise in KGs.

Acknowledgment

This work was funded by Science Research Project of Hebei Education Department (QN2024115), the Fundamental Research Funds for the Hebei University of Water Resources and Electirc Engineering (SYKY2311,SYKY2310).

References

[1] Z. Sun, Q. Guo, J. Yang, H. Fang, G. Guo, J. Zhang, and R. Burke, (2019) *"Research Commentary on Recommendations with Side Information: A Survey and Research Directions"* **Electronic Commerce Research and Applications** ***37***: 100879. DOI: 10.1016/j.elerap.2019.100879.

[2] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua. "Explainable Reasoning over Knowledge Graphs for Recommendation". In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. 2019, 5329–5336. DOI: 10.1609/aaai.v33i01.33015329.

[3] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu. "Recurrent Knowledge Graph Embedding for Effective Recommendation". In: *Proceedings of the 12th ACM Conference on Recommender Systems*. 2018, 297–305. DOI: 10.1145/3240323.3240361.

[4] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, (2018) *"Heterogeneous Information Network Embedding for Recommendation"* **IEEE Transactions on Knowledge and Data Engineering** ***31***(2): 357–370. DOI: 10.1109/TKDE.2018.2833443.

[5] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo. "Knowledge Graph Convolutional Networks for Recommender Systems". In: *Proceedings of the 28th World Wide Web Conference*. 2019, 3307–3313. DOI: 10.1145/3308558.3313417.

[6] Y. Chen, Y. Yang, Y. Wang, J. Bai, X. Song, and I. King. "Attentive Knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation". In: *2022 IEEE 38th Interna-*

tional Conference on Data Engineering (ICDE). IEEE. 2022, 299–311. DOI: 10.1109/ICDE53745.2022.00027.

[7] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang. "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discover and Data Mining*. 2019, 968–977. DOI: 10.1145/3292500.3330836.

[8] X. Sha, J. Wang, J. Ding, and X. Xu, (2024) *"Hierarchical Knowledge graph embedding with self-attention mechanism for recommendation"* **Journal of Frontiers of Computer Science and Technology**: 1–15. DOI: 10.3778/j.issn.1673-9418.2407019.

[9] Z. Liu, V. W. Zheng, Z. Zhao, Z. Li, H. Yang, M. Wu, and J. Ying. "Interactive Paths Embedding for Semantic Proximity Search on Heterogeneous Graphs". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2018, 1860–1869. DOI: 10.1145/3219819.3219953.

[10] X. Xin, X. He, Y. Zhang, Y. Zhang, and J. Jose. "Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, 125–134. DOI: 10.1145/3331184.3331188.

[11] 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: *Proceedings of the 28th World Wide Web Conference*. 2019, 151–161. DOI: 10.1145/3308558.3313705.

[12] C. Chen, M. Zhang, W. Ma, Y. Liu, and S. Ma. "Jointly Non-sampling Learning for Knowledge Graph Enhanced Recommendation". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020, 189–198. DOI: 10.1145/3397271.3401040.

[13] P. Wang, Y. Fan, L. Xia, W. X. Zhao, S. Niu, and J. Huang. "KERL: A Knowledge-Guided Reinforcement Learning Model for Sequential Recommendation". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020, 209–218. DOI: 10.1145/3397271.3401134.

[14] C. Wang, M. Zhang, W. Ma, Y. Liu, and S. Ma. "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*. 2020, 109–118. DOI: 10.1145/3397271.3401131.

[15] Z. Lin and L. Qu, (2023) *"Collaborative Knowledge-aware Recommendation based on Neighborhood Negative Sampling"* **Information Systems *115***: 102207. DOI: 10.1016/j.is.2023.102207.

[16] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu. "Heterogeneous Graph Attention Network". In: *Proceedings of the 28th International Conference on World Wide Web*. 2019, 2022–2032. DOI: 10.1145/3308558.3313562.

[17] Y. Lu, Y. Fang, and C. Shi. "Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2020, 1563–1573. DOI: 10.1145/3394486.3403207.

[18] K. Yuan, G. Liu, J. Wu, and H. Xiong, (2023) *"Semantic and Structural View Fusion Modeling for Social Recommendation"* **IEEE Transactions on Knowledge and Data Engineering *35***(11): 11872–11884. DOI: 10.1109/TKDE.2022.3230972.

[19] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, and T.-S. Chua. "Learning Intents Behind Interactions with Knowledge Graph for Recommendation". In: *Proceedings of the Web Conference 2021*. 2021, 878–887. DOI: 10.1145/3442381.3450133.

[20] Y. Li, X. Sun, H. Chen, S. Zhang, Y. Yang, and G. Xu, (2024) *"Attention Is Not the Only Choice: Counterfactual Reasoning for Path-based Explainable Recommendation"* **IEEE Transactions on Knowledge and Data Engineering**: DOI: 10.1109/TKDE.2024.3373608.

[21] J. Jin, J. Qin, Y. Fang, K. Du, W. Zhang, Y. Yu, Z. Zhang, and A. J. Smola. "An Efficient Neighborhood-based Interaction Model for Recommendation on Heterogeneous Graph". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2020, 75–84. DOI: 10.1145/3394486.3403050.

[22] X. Chen, C. Huang, L. Yao, X. Wang, W. Zhang, et al. "Knowledge-guided Deep Reinforcement Learning for Interactive Recommendation". In: *2020 International Joint Conference on Neural Networks*. IEEE. 2020, 1–8. DOI: 10.1109/IJCNN48605.2020.9207010.

[23] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua. "KGAT: Knowledge Graph Attention Network for Recommendation". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, 950–958. DOI: 10.1145/3292500.3330989.

[24] Y. Yang, C. Huang, L. Xia, and C. Li. "Knowledge Graph Contrastive Learning for Recommendation". In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2022, 1434–1443. DOI: 10.1145/3477495.3532009.

[25] W. Zhang, L. Yang, Z. Song, H. P. Zou, K. Xu, L. Fang, and P. S. Yu. "Do We Really Need Graph Convolution During Training? Light Post-Training Graph-ODE for Efficient Recommendation". In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2024, 3248–3258. DOI: 10.1145/3627673.3679773.

[26] B. Shuai, Z. Zuo, B. Wang, and G. Wang. "Dag-recurrent Neural Networks for Scene Labeling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, 3620–3629. DOI: 10.1109/CVPR.2016.394.

[27] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. "Neural Collaborative Filtering". In: *Proceedings of the 26th International Conference on World Wide Web*. 2017, 173–182. DOI: 10.1145/3038912.3052569.

[28] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. "Learning Entity and Relation Embeddings for Knowledge Graph Completion". In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2015, 2181–2187. DOI: 10.1609/aaai.v29i1.9491.

[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *NeurIPS 2014 Workshop on Deep Learning*. 2014. DOI: https://doi.org/10.48550/arXiv.1412.3555.

[30] 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: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2018, 1531–1540. DOI: 10.1145/3219819.3219965.

[31] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. "Personalized Entity Recommendation: A Heterogeneous Information Network Approach". In: *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. 2014, 283–292. DOI: 10.1145/2556195.2556259.

[32] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. "BPR: Bayesian Personalized Ranking from Implicit Feedback". In: *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. 2009, 452–461. DOI: 10.5555/1795114.1795167.

[33] G. Lee, K. Kim, and K. Shin. "Revisiting LightGCN: Unexpected Inflexibility, Inconsistency, and A Remedy Towards Improved Recommendation". In: *Proceedings of the 18th ACM Conference on Recommender Systems*. 2024, 957–962. DOI: 10.1145/3640457.3688176.

[34] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma. "Collaborative Knowledge Base Embedding for Recommender Systems". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, 353–362. DOI: 10.1145/2939672.2939673.

[35] Y. Jiang, Y. Yang, L. Xia, and C. Huang. "DiffKG: Knowledge Graph Diffusion Model for Recommendation". In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 2024, 313–321. DOI: 10.1145/3616855.3635850.

[36] D. P. Kingma and J. Ba, (2014) *"Adam: A Method for Stochastic Optimization"* **arXiv preprint**: arXiv:1412.6980. DOI: https://arxiv.org/abs/1412.6980.

[37] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua. "Neural Graph Collaborative Filtering". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, 165–174. DOI: 10.1145/3331184.3331267.

[38] L. Zimmer, M. Lindauer, and F. Hutter, (2021) *"Auto-Pytorch: Multi-fidelity Metalearning for Efficient and Robust Autodl"* **IEEE Transactions on Pattern Analysis and Machine Intelligence** *43*(9): 3079–3090. DOI: 10.1109/TPAMI.2021.3067763.

[39] T. Zhao, J. Chen, Y. Ru, Q. Lin, Y. Geng, and J. Liu. "Untargeted Adversarial Attack on Knowledge Graph Embeddings". In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024, 1701–1711. DOI: 10.1145/3626772.3657702.

[40] Z. Guo, P. Song, C. Feng, K. Yao, C. Dang, and J. Liang, (2024) *"Causal Intervention for Knowledge Graph Denoising in Recommender Systems"* **International Journal of Machine Learning and Cybernetics**: 1–17. DOI: 10.1007/s13042-024-02500-0.