

Article

A Recommendation Algorithm Combining Local and Global Interest Features

Xiaoyuan Song ^{1,2}, Jiwei Qin ^{1,2,*} and Qiulin Ren ^{1,2}¹ School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China² Key Laboratory of Signal Detection and Processing, Xinjiang Uygur Autonomous Region, Xinjiang University, Urumqi 830046, China

* Correspondence: jwqin_xju@163.com

Abstract: Due to the ability of knowledge graph to effectively solve the sparsity problem of collaborative filtering, knowledge graph (KG) has been widely studied and applied as auxiliary information in the field of recommendation systems. However, existing KG-based recommendation methods mainly focus on learning its representation from the neighborhood of target items, ignoring the influence of other items on the target item. The learning focuses on the local feature representation of the target item, which is not sufficient to effectively explore the user's preference degree for the target item. To address the above issues, in this paper, an approach combining users' local interest features with global interest features (KGG) is proposed to efficiently explore the user's preference level for the target item, which learns the user's local interest features and global interest features for target item through Knowledge Graph Convolutional Network and Generative Adversarial Network (GAN). Specifically, this paper first utilizes the Knowledge Graph Convolutional Network to mine related attributes on the knowledge graph to effectively capture item correlations and obtain the local feature representation of the target item, then uses the matrix factorization method to learn the user's local interest features for target items. Secondly, it uses GAN to learn the user's global interest features for target items from the implicit interaction matrix. Finally, a linear fusion layer is designed to effectively fuse the user's local and global interests towards target items to obtain the final click prediction. Experimental results on three real datasets show that the proposed method not only effectively integrates the user's local and global interests but also further alleviates the problem of data sparsity. Compared with the current baselines for knowledge graph-based systems, the KGG method achieves a maximum improvement of 8.1% and 7.6% in AUC and ACC, respectively.

Keywords: recommendation system; knowledge graph; Knowledge Graph Convolutional Network; Generative Adversarial Network; interest features



Citation: Song, X.; Qin, J.; Ren, Q. A Recommendation Algorithm Combining Local and Global Interest Features. *Electronics* **2023**, *12*, 1857. <https://doi.org/10.3390/electronics12081857>

Academic Editor: George A. Tsihrintzis

Received: 6 March 2023

Revised: 26 March 2023

Accepted: 7 April 2023

Published: 14 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the continuous development of Internet technology, recommendation systems have been widely used in e-commerce, social networks, music and news to solve the problem of information explosion. Among them, collaborative filtering (CF) based recommendation algorithms are one of the most widely used and popular technologies [1]. Collaborative filtering [2–4] infers unknown data through historical data of user interaction with items to recommend items that users may be interested in. However, collaborative filtering-based recommendation algorithms usually have data sparsity and cold start problems. Therefore, many researchers have proposed using knowledge graphs [5–9] as auxiliary information to alleviate the problem of data sparsity.

A knowledge graph is a commonly used auxiliary information in recommendation systems, which usually contains richer facts and connections about items. A knowledge graph is a heterogeneous graph, mainly composed of triples (entity, relation, entity). Among them, nodes correspond to entities, and edges correspond to relations. The recommendation

algorithm based on a knowledge graph uses the rich semantic association between items to improve the performance of the recommendation system. Specifically, it aggregates neighbor nodes around target items in knowledge graphs to enrich the representation vector of target items. At present, knowledge graphs have been successfully applied to recommendation systems, and researchers have studied many classic algorithms.

PER [10] is a path-based recommendation algorithm that views KG as a heterogeneous information network and extracts latent features between users and items from manually constructed entity-entity paths. However, PER relies on path design, which is difficult to construct reasonable entity paths from massive data manually. CKE [11] combines the CF module with knowledge embedding, text embedding and item image embedding in a unified Bayesian framework to improve the performance of the recommendation system, but it does not consider the differences in different modal features in information fusion. DKN [12] combines news title embedding with entity embedding in the knowledge graph to enhance news feature representation. However, DKN is not an end-to-end training and requires pre-acquisition of entity embedding representations, which increases model complexity. RippleNet [13] views the user's historical interests as seed sets in KG and iteratively expands the user's interests along KG links to discover his latent interest in candidate items. However, when exploring users' interests along KG links, authors ignore the influence of relations on user interests. MKR [14] shares high-order interactive representations learned from KG through multi-task learning to supplement the insufficient item representation caused by data sparsity. KGNN-LS [15] transforms the knowledge graph into a user-specific weighted graph using label smoothness regularization and then applies graph neural networks to calculate personalized project embedding representations. Finally, the user's click probability is predicted through the inner product. KGCN [16] aggregates neighborhood information to obtain high-order structural information and semantic information from KG to enrich target entity representation and improve recommendation performance. KGAT [17] combines an embedding propagation layer with an attention mechanism, which adaptively propagates embeddings from the neighbors of the target item to update its representation, ultimately capturing high-order structural information based on both user behavior and item attributes and efficiently utilizing rich information in the knowledge graph. CKAN [18] explicitly encodes collaborative signals through collaborative propagation and combines them with knowledge associations to enrich user and item embeddings. In the propagation process, it selects neighbors through a knowledge-aware attention mechanism and aggregates the representations of users and items from different propagation layers as input to the prediction layer, predicting the preference score of the user for the item. KGIN [19] proposes an intent network based on the knowledge graph, which identifies important intents and relation paths to characterize the correlation between users and items, providing interpretability and better recommendation performance for prediction. CKEN [20] encodes the user-item collaboration information explicitly using a collaborative propagation method and propagates it in the knowledge graph to obtain item representations. In addition, it designs a feature interaction layer to enable feature sharing between the items in the recommendation system and the entities in the knowledge graph, further enriching the latent vector representations of the items.

However, the above methods, although focusing on the impact of entities and relationships on users when obtaining rich item representations through knowledge graphs, are limited by the size of the receptive field, making it difficult to involve every neighboring node and lacking the exploration of users' global interest characteristics, which is insufficient to effectively explore users' interest distribution.

In order to better capture users' local and global interests for a recommendation, this paper proposes a recommendation algorithm combining local interest features and global interest features, which integrates knowledge graph information and rating information, called KGG. Figure 1 is the framework diagram of KGG. KGG consists of three parts: local feature learning module, global feature learning module and linear fusion module. First, in the local feature learning module, we used KGCN to aggregate the node information in the

neighborhood to obtain the feature representation of the target item, then calculated the dot product between the obtained item feature vector and user feature vector to obtain local interactive features. Secondly, in the global feature learning module, we used Generative Adversarial Network (GAN) to learn user interest distribution from the rating matrix and finally obtained global interactive features. Finally, we used a linear fusion module to unify local features and global features to predict the user's click probability.

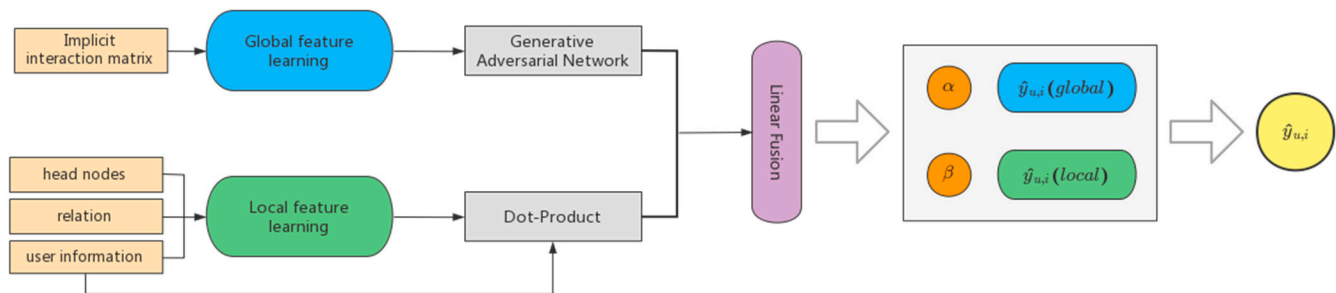


Figure 1. KGG model framework diagram.

We conducted extensive experiments on three real-world datasets to evaluate the performance of our model. The experimental results validated the complementarity of two feature learning modules and produced better prediction performance when fused. In addition, our proposed KGG significantly outperformed existing baselines in all three datasets. In summary, the contributions of this paper are as follows:

1. We combined the Knowledge Graph Convolutional Network and Generative Adversarial Network organically, and our model can learn both local and global interests effectively to explore users' interest preferences.
2. By utilizing rating information and knowledge graph information, we can further alleviate the data sparsity problem existing in recommendation systems.
3. Through extensive experiments on three real-world datasets, the results show that our proposed KGG achieves better click prediction accuracy than current state-of-the-art methods.

The rest of the paper is organized as follows: Section 2 summarizes the related work. Section 3 provides the problem definition and proposed model. Section 4 analyzes the experimental results. In Section 5, we conclude this paper.

2. Related Work

2.1. Recommendation Method Based on Generative Adversarial Network

In recent years, Generative Adversarial Networks (GANs) [21–23] have been applied to the recommendation field due to their strong ability to learn complex data distributions. With the continuous development of GANs, Jun Wang et al. first applied mature GAN models to the recommendation field in 2017 and proposed the IRGAN model [24], which uses GANs to learn user's interest distribution and combines collaborative filtering to predict how interested users are in projects. Dong-Kyu Chae et al. proposed the CFGAN model [25], which uses real-valued vectors for training, thus fully utilizing GAN's adversarial characteristics. Y. Tong et al. proposed the CGAN model [26], which replaces the generator in GAN with a variational autoencoder [27], transforming adversarial training from discrete space to continuous space, thus improving the robustness of the model. Yuan Lin et al. proposed the IFGAN model [28], which further introduces an additional generator into the original GAN, increasing information flow between generators and reducing differences between generators and discriminators.

2.2. Recommendation Method Based on Knowledge Graph

Currently, recommendation methods based on knowledge graphs can be divided into three categories: (1) Embedding-based methods [29–34], which obtain rich representations

of entities through knowledge graph embedding and then use them for the recommendation. Embedding-based methods have high flexibility in the auxiliary recommendation, but they are more suitable for graph applications due to their focus on modeling semantic relevance. (2) Path-based methods [35–39], which assist in recommendations by exploring various connection patterns between items in the knowledge graph. Although path-based methods can intuitively use KG, the design of paths often requires manual work, which is time consuming and laborious in practice. (3) Hybrid methods [40–43], which combine embedding-based methods and path-based methods to obtain item embedding representations.

3. Method

In this section, we introduce the proposed KGG model. First, we define the research question. Then, we introduce the various modules of KGG in detail. Finally, we present the complete learning algorithm of KGG.

3.1. Problem Definition

We assumed that there are M groups of users $U = \{u_1, u_2, \dots, u_m\}$ and N groups of items $I = \{i_1, i_2, \dots, i_n\}$ in a particular scenario.

In this paper, we used an implicit interaction matrix $Y \in R^{M \times N}$, where if a user has rated or interacted with a certain item, the corresponding position of the item in the implicit interaction matrix is set to 1 ($y_{ui} = 1$), and if not rated or interacted, the value at that position is 0 ($y_{ui} = 0$). Additionally, we used a knowledge graph G containing head entities, relations and tail entities (h, r, t) , where $h \in H, r \in R, t \in H, H$ is the set of entities, and R is the set of relations. For example, triples (Full River Red, movie. movie. director, Zhang Yimou) indicates that Full River Red was directed by Zhang Yimou. In many recommendation application scenarios, an item $i \in I$ also corresponds to an entity $e \in E$. As in the above example, Full River Red also corresponds to an entity in the knowledge graph.

In this paper, we used GAN to learn the global interest features of users in the implicit interaction matrix Y and used the KGCN method to learn the local interest features of users in the knowledge graph. Our ultimate goal is to predict whether there is a potential interest for user u on an uninteracted item i , namely $\hat{y}_{ui} = F(u, i | \theta, Y, G)$, where θ represents the model parameters of function F .

3.2. Local Feature Learning Module

In the local feature learning module, this paper mainly refers to the KGCN [16] method proposed by Hongwei Wang et al. KGCN is a graph-based deep learning framework that captures the relationship between entities by mapping node and edge features to a low-dimensional space to obtain the feature representation of the target item. In this section, we focus on introducing the principle of KGCN.

In the KGCN layer, different users have different levels of interest in different relationships.

For example, u_1 loves Zhang Yimou very much, so he gives a high score to the movie Manchurian Red, while u_2 only likes love movies and is not interested in who directed it, so he will not watch Manchurian Red or give it a low score after watching. Therefore, the KGCN model needs to calculate the degree of users' interest in each relationship, namely, the scores, and normalize all the relationship scores to obtain the probability of users' possible liking for each relationship as weights for neighbor node representation.

The formula for calculating the score between a user and a relationship is as follows:

$$S_r^u = d(u, r) \quad (1)$$

S represents the score of the user and the relationship, d represents the dot product calculation, and the dimension size of user u and relationship r is the same.

After calculating the scores of each relationship for the user, all scores are normalized to obtain the final probability of the user's interest in each relation, as shown in Formula (2):

$$\tilde{S}_{r_{i,e}}^u = \frac{\exp(S_{r_{i,e}}^u)}{\sum_{e \in N(i)} \exp(S_{r_{i,e}}^u)} \quad (2)$$

where $r_{i,e}$ represents the relationship between entity i and entity e , $S_{r_{i,e}}^u$ represents the score of user and relation, $N(i)$ represents the set of entities directly connected to item i . As shown in Figure 2, red nodes represent our target items; blue nodes represent our neighbor nodes. If the field size is 1 ($D = 1$), there are three entities directly connected to the red node e_1, e_2, e_3 , then $N(i) = \{e_1, e_2, e_3\}$.

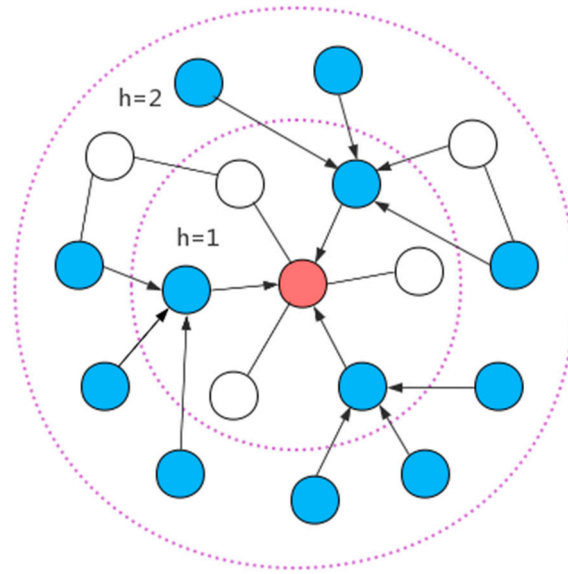


Figure 2. KGCN Model Diagram: the red circle represents the target project, the blue circles represent the extracted neighboring nodes, and the white circles represent the unextracted neighboring nodes.

The probability of the user being interested in the relationship is calculated as the weight represented by the neighboring node, which is specifically calculated according to Formula (3):

$$i_{N(i)}^u = \sum_{e \in N(i)} \tilde{S}_{r_{i,e}}^u \cdot e \quad (3)$$

If the receptive field size is 1 ($D = 1$) in Figure 2, then the representation of the neighboring nodes is $v = \tilde{S}_{r_{1,e_1}}^u \cdot e_1 + \tilde{S}_{r_{2,e_2}}^u \cdot e_2 + \tilde{S}_{r_{3,e_3}}^u \cdot e_3$.

The last step of the KGCN layer is to represent the neighboring nodes and the target item as a single vector in an aggregation manner. There are three main aggregation methods: Sum aggregator, Concat aggregator and Neighbor aggregator. In this model application, we mainly use the Sum aggregator. The calculation method is shown in Formula (4):

$$agg_{sum} = \sigma(W \cdot (i + i_{N(i)}^u) + b) \quad (4)$$

where i is the vector representation of the target item itself, and $i_{N(i)}^u$ is the vector representation of its neighbor node.

Finally, we predict the target items by using the target item feature representations learned from the KGCN layer and user representations as shown in Equation (5):

$$y_{ui}(local) = f(u, i^u) \quad (5)$$

where \hat{y}_{ui} represents the local interest prediction, i^u is the feature vector of item i corresponding to user u , and f is the dot product operation.

In the local feature learning module, we use the cross-entropy loss for training, which is calculated as shown in Formula (6):

$$L(local) = \sum_{u \in U} (\sum_{i: y_{ui}=1} C(y_{ui} - \hat{y}_{ui})) + \lambda \cdot \|F\|_2^2 \quad (6)$$

C represents cross-entropy loss, and F represents the $L2$ regularization term.

3.3. Global Feature Learning Module

In the global feature learning module, the CFGAN model framework was used in this paper. In the training schemes proposed by the previous researchers, there was no consideration of the generated samples being discrete item indices, which easily leads to Discriminator (D) being unable to distinguish the source of input and thus cannot fully leverage adversarial training to learn user's interest distribution. To address this issue, Dong-Kyu Chae et al. proposed the CFGAN model [25]. CFGAN uses G and D for adversarial training, where G tries to generate continuous real-valued element purchase vectors instead of discrete item indices, and D tries to distinguish generated real-valued purchase vectors from real ones. This can avoid the confusion of D caused by mutually contradictory labels in current GAN-based recommendation algorithms and make the discriminator guide G through back propagation to continuously improve so that generated vectors become closer to actual purchase vectors during the iteration process. Finally, a higher precision recommendation can be achieved.

In this paper, we only introduced the key part of CFGAN. Figure 3 is the framework of the CFGAN model, which has specific user conditions, that is, to learn the parameters of the model under the condition of considering user personalization. Given a specific user condition vector c_u (user attribute features) and real user purchase vector r_u (position interacted with users and items is 1, otherwise 0). The purpose of G is to generate vectors that are as close as possible to real purchase vectors, while D's goal is to distinguish input vectors from generators or discriminators as much as possible.

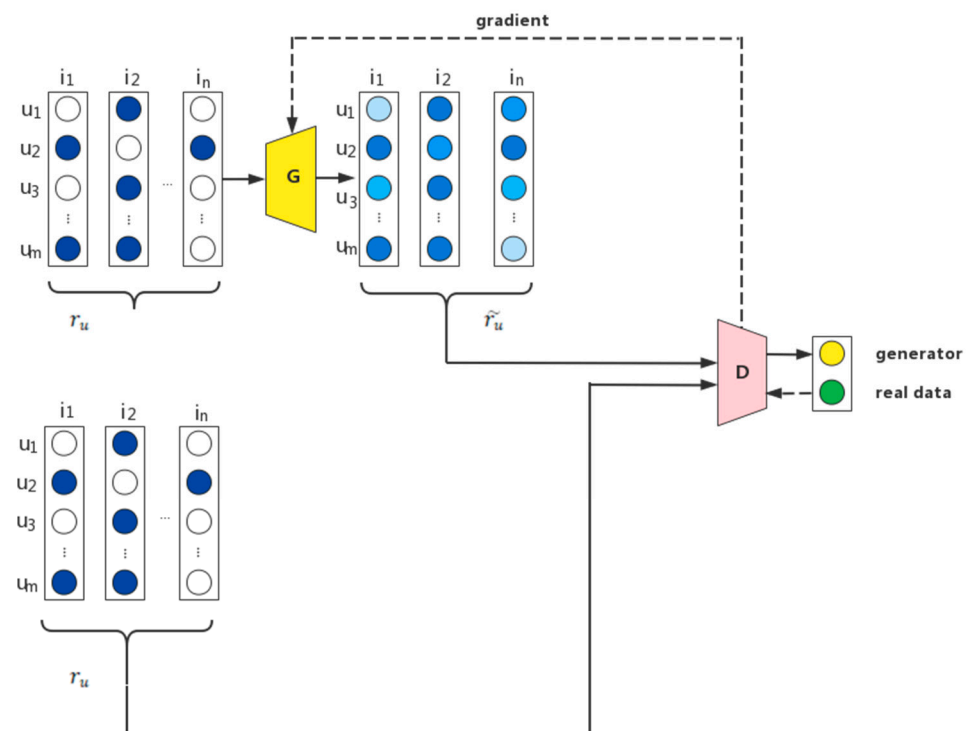


Figure 3. Framework diagram of CFGAN model.

Formally, the D's objective function, denoted as J^D , is as follows:

$$J^D = -\sum_u (\log D(r_u | c_u) + \log(1 - D(r_u \odot (e_u + k_u) | c_u))) \quad (7)$$

where r_u is the true purchase vector, \hat{r}_u is the generated purchase vector, \odot is the dot product operation, e_u is an indicator vector (if u purchased i , then $e_{ui} = 1$; otherwise $e_{ui} = 0$), and k_u is an indicator vector (if j belongs to the negative sample set sampled in training, then $k_{uj} = 1$; otherwise $k_{uj} = 0$).

The loss function of G is shown in Formula (8):

$$J^G = \sum_u \log(1 - D(\hat{r}_u \odot (e_u + k_u) | c_u)) + \gamma \sum_j (x_{uj} - \hat{x}_{uj}) \quad (8)$$

Among them, γ is the second regularization term, j is the weight parameter, and j is the non-interaction item.

For D's loss function J^D , the main goal is that if the input vector comes from the generator, it is better for its output value to be close to 0 after passing through D; if the input vector comes from a sampled real purchase vector, it is better for its output value to be close to 1 after passing through D, and D reversely transmits the obtained information to G and guide G's training. For G's loss function J^G , the main goal is that after passing through D, the output value of the G-generated purchase vector is as close to 1 as possible, namely, as close as possible to real purchase vectors.

After G and D underwent continuous confrontation training, they eventually reached a balanced point where D could not distinguish the source of the input. At this point, the model training was completed. The dense vectors generated by G serve as scores for each user on what they may like, and by sorting these scores and selecting the top N items to recommend to the user, the predicted result is presented $\hat{y}_{ui}(global)$.

3.4. Linear Fusion Module

The above two modules capture the local and global interest features of users from different sources (rating matrix and knowledge graph), respectively. This paper unifies the local and global together through a designed linear fusion module, whose specific calculation method is shown in Formula (9):

$$\hat{y}_{ui} = \sigma(\beta \cdot \hat{y}_{ui}(local) + \alpha \cdot \hat{y}_{ui}(global)) \quad (9)$$

where the sigmoid function is used to limit our prediction value between 0 and 1, and β and α are weights of local and global interest features. Different proportions of local and global interest features have different impacts on the accuracy of our prediction. In the experimental part, we analyzed these two hyperparameters.

3.5. KGG Model

The KGG model adopts a step-wise training approach, where the local feature learning module and global feature learning module are independent of each other and trained separately. Finally, the linear fusion module is used to obtain the final click prediction result.

4. Experiment

We conducted experiments and evaluations on three real-world datasets with our proposed KGG model. Our main research questions are as follows:

- (1) How does our KGG model compare to existing knowledge graph-based recommendation models?
- (2) What is the impact of the hyperparameters of the linear fusion module on our model?
- (3) Is it effective to combine the local interest feature learning module and the global interest feature learning module?

4.1. Datasets

In order to verify the performance of the KGG model, we used three common datasets: Last.FM, Movielens-1M and Book-Crossing. These datasets all contain user and item rating histories, which we processed as implicit ratings where the position of items rated is 1 and 0 otherwise. For each dataset, we randomly selected 60% of the user interactions as a training set, 20% as a test set and 20% as a validation set. The information on the datasets is shown in Table 1.

Table 1. Information of datasets.

Datasets	Last.FM	Movielens-1M	Book-Crossing
users	1872	6036	17,860
items	3846	2347	14,910
interactions	42,346	753,772	139,746
entities	9366	7008	24,127
relations	60	7	10
KG triples	15,518	20,782	19,793

Last.FM: A commonly used music-type dataset in recommendation systems, containing 1872 users, 3846 items, 42,346 interactions histories, 9366 entities and 15,518 triples.

Movielens-1M: A classic movie-type dataset with a large amount of user interaction data and widely used, containing 6036 users, 2347 items, 753,772 interactions histories, 7008 entities and 20,782 triples.

Book-Crossing: A large shared book dataset widely used in the recommendation. It contains 17,860 users, 14,910 items, 139,746 interactions histories, 24,127 entities and 19,793 triples.

4.2. Baselines

We compared our proposed KGG model with the following baselines, including one without KG and five based on KG:

PER [10]: Viewing KG as a heterogeneous graph and exploring the various relationships between projects through path-based methods.

CKE [11]: An integrated recommendation framework that combines collaborative filtering with structural, textual, and visual knowledge.

DKN [12]: Using CNN to combine entity embedding and word embedding as multiple channels for click prediction.

Wide&Deep [44]: Combines a generalized linear model with embeddings and multi-layer perceptrons from deep learning to predict clicks.

RippleNet [13]: A memory-network-like approach is employed to propagate users' preferences on the KG for the recommendation.

KGCN [16]: Utilize the neighbor nodes of the target item in the knowledge graph reasonably to enrich the feature representation of the target item and explore users' personalized and latent interests.

KGAT [17]: A recommendation algorithm that adaptively obtains the embedding representation of the target item from its neighboring nodes through an attention mechanism.

KGIN [19]: A knowledge graph-based intent network is proposed, which utilizes users' intents and relation paths to characterize the correlation between users and items.

4.3. Experiments Setup

In the click-through rate prediction, AUC (area under the curve) and ACC [45] are selected as evaluation indexes to measure the performance of KGG prediction. The calculation method for AUC is as follows:

$$AUC = \frac{\sum I(P_{positive}, P_{negative})}{M \times N} \quad (10)$$

Among,

$$I(P_{positive}, P_{negative}) = \begin{cases} 1, P_{positive} > P_{negative} \\ 0.5, P_{positive} = P_{negative} \\ 0, P_{positive} < P_{negative} \end{cases} \quad (11)$$

M represents the number of positive samples, and N represents the number of negative samples; there are a total of $M \times N$ pairs of samples (one pair of samples consists of one positive sample and one negative sample). Count the number of pairs in which the predicted probability of the positive sample is greater than that of the negative sample among all the samples.

The calculation method for ACC is as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

where TP represents True Positive, TN represents True Negative, FP represents False Positive, and FN represents False Negative.

In the KGG model, the local feature learning module and global feature learning modules are trained step-by-step, so two parts of experimental parameters are involved. The specific parameter settings are as follows:

In the local interest feature learning module, different parameters are set for different datasets. In Last.FM, the entity embedding dimension d is 2048, the neighbor node size K is 8, the receptive field depth D is 1, the epoch is 20, the batchsize is 1024, the learning rate of $l2$ regularization λ is 10^{-7} , and the model training learning rate η is 10^{-3} . In Movielens-1M, the entity embedding dimension d is 512, neighbor node size K is 8, receptive field depth D is 1, the epoch is 20, the batchsize is 256, $l2$ regularization learning rate λ is 10^{-7} , and model training learning rate η is 10^{-3} . In Book-Crossing, the entity embedding dimension d is 256, the neighbor node size K is 8, the receptive field depth D is 3, the epoch is 20, the batchsize is 20, $l2$ regularization learning rate λ is 10^{-5} , and the model training learning rate η is 10^{-3} .

In the Global Interest Feature Learning module, the network layers of the generator and encoder are 4, the hidden layer size is 400, the regularization weight is 0.1, and the learning rate is 0.001. In Last.FM, epochs are 1000 and batchsize is 32. In Movielens-1M, epochs are 1500 and batchsize is 64. In Book-Crossing, epochs are 1500 and batchsize is 256.

In the linear fusion module, the range of β and α selection for hyperparameters is {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.

4.4. Results

4.4.1. Comparison with the Baselines (RQ1)

In this section, we compared KGG with the baselines proposed in Section 4.2, and the experimental results are shown in Table 2. Intuitively, our proposed KGG model has made great improvements compared to the existing baselines. We draw the following conclusion from the experimental results:

Table 2. Comparison results.

Model	Last.FM		Movielens-1M		Book-Crossing	
	AUC	ACC	AUC	ACC	AUC	ACC
PER	0.633	0.596	0.710	0.664	0.623	0.588
CKE	0.744	0.673	0.801	0.742	0.671	0.633
DKN	0.602	0.581	0.655	0.589	0.622	0.598
Wide&Deep	0.756	0.688	0.898	0.820	0.712	0.624
RippleNet	0.768	0.691	0.920	0.842	0.729	0.662
KGCN	0.790	0.702	0.919	0.842	0.672	0.617
KGAT	0.716	0.647	0.901	0.828	0.651	0.624
KGIN	0.818	0.742	0.919	0.844	0.727	0.660
Ours	0.823	0.751	0.926	0.852	0.731	0.668

In Last.FM, our proposed KGG method improves AUC by 4.0% and ACC by 6.5% compared with KGCN. In Movielens-1M, AUC improves by 0.7%, and ACC improves by 1.2%. In Book-Crossing, AUC improves by 8.1%, and ACC improves by 7.6. Although RippleNet, DKN, CKE, and PER can capture the rich semantic relationships among items from the knowledge graph and improve the interpretability of recommendations, these knowledge graph-based approaches suffer from certain shortcomings and cannot capture the global interest distribution of users. While KGG compares with them, the maximum improvement of AUC is 23.1%, and ACC is 20.6% in Last.FM, 23.3% and 22.1% in Movielens-1M, and 14.9% and 10.5% in Book-Crossing. The results show that compared with knowledge graph-based methods, KGG combines local interest features with global interest features to fully explore the overall interest distribution of users, improving the accuracy of the recommended model.

Wide&Deep is a classical combined linear and nonlinear model which synthesizes the linear and nonlinear features of users and can be regarded as a model for exploring the global interest distribution of users. Compared with Wide&Deep, in Last.FM, the AUC is improved by 8.1% and ACC by 8.4%; in Movielens-1M, AUC is improved by 3.0% and ACC by 3.8%; in Book-Crossing, the AUC is improved by 2.6% and ACC by 6.6%. The advantage of KGG lies in that it uses local and global interest features extracted by a neural network to obtain complete user interest features through linear fusion at the same time of linear and nonlinear combination; at the same time, we are able to use more information to explore the distribution of users' interests effectively.

Compared with the aforementioned methods on three datasets, KGAT and KGIN achieve better recommendation performance, and the attention mechanism and user intent in KGAT and KGIN can further explore the rich information in the knowledge graph, indicating that different neighbor nodes and relationships have varying degrees of importance for users. However, KGG outperforms KGAT and KGIN, indicating that local interest features alone are insufficient to represent the overall interests of users, and the fusion of local and global interest features can more effectively explore the interest distribution of users.

From the evaluation metrics, KGG shows excellent performance on Last.FM, Book-Crossing and Movielens-1M. On the datasets, Last.FM and Book-Crossing are much more sparse than Movielens-1M, while KGG's improvement on Last.FM and Book-Crossing is higher than that of Movielens-1M. We believe that KGG, which combines local and global information, can not only comprehensively explore users' interests and preferences but also alleviate the existing data sparsity problem to improve the accuracy of click rate prediction.

4.4.2. Hyperparameter Analysis (RQ2)

In the analysis and experiment, we set the other parameters of the model, as shown in Section 4.4.1. Figures 4–6 show the influence of β and α on model performance under Last.FM, Movielens-1M and Book-Crossing, respectively, where the abscissa indicates the values of β and α . From these three figures, it can be seen that with the increasing of β and α , the accuracy of click rate prediction is higher in different datasets. When β is 0.1 and α is 0.9, the performance of the model reaches its maximum. We will analyze cases with $\beta = 0, \alpha = 1$ or $\beta = 1, \alpha = 0$ in ablation experiments.

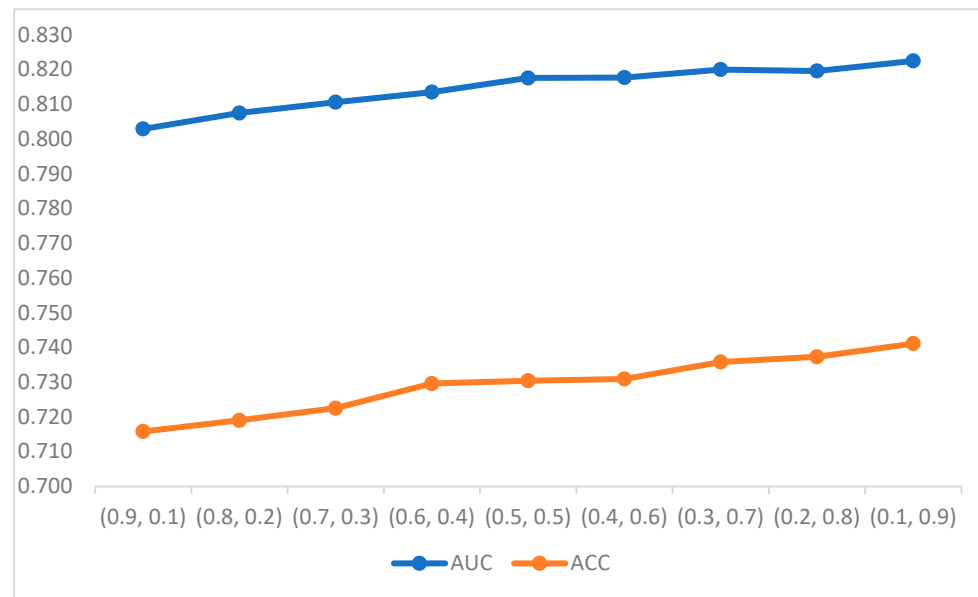


Figure 4. Impact of different hyperparameters on the performance of KGG model on Last.FM.

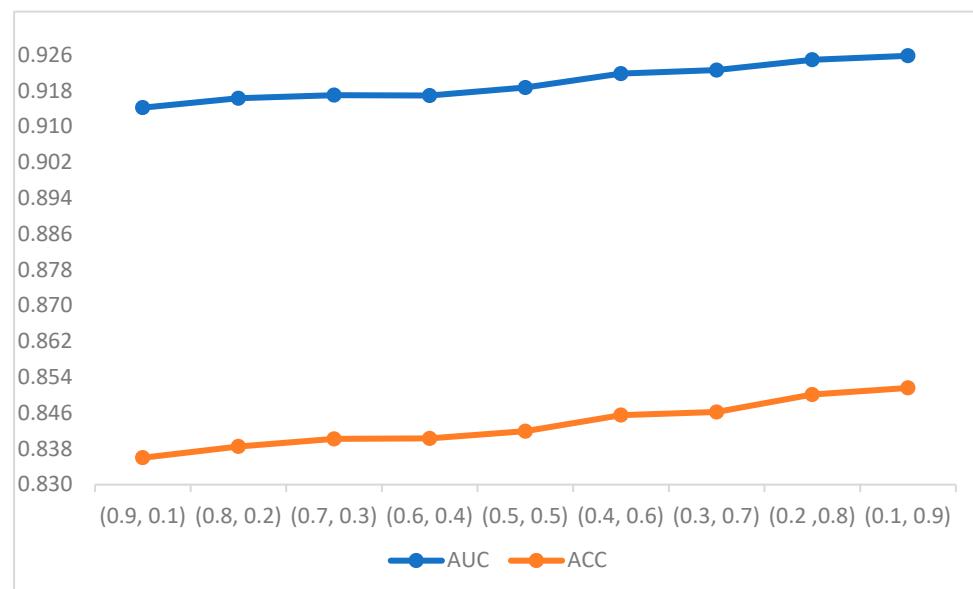


Figure 5. Impact of different hyperparameters on the performance of KGG model on Movielens-1M.

From the experimental results shown in Figures 4–6, we can discover that as the weight of global interest features increases, the accuracy of KGG model prediction also increases. This explains to some extent the importance of global interest features in the KGG model and plays a key role in effectively exploring user interest distribution.

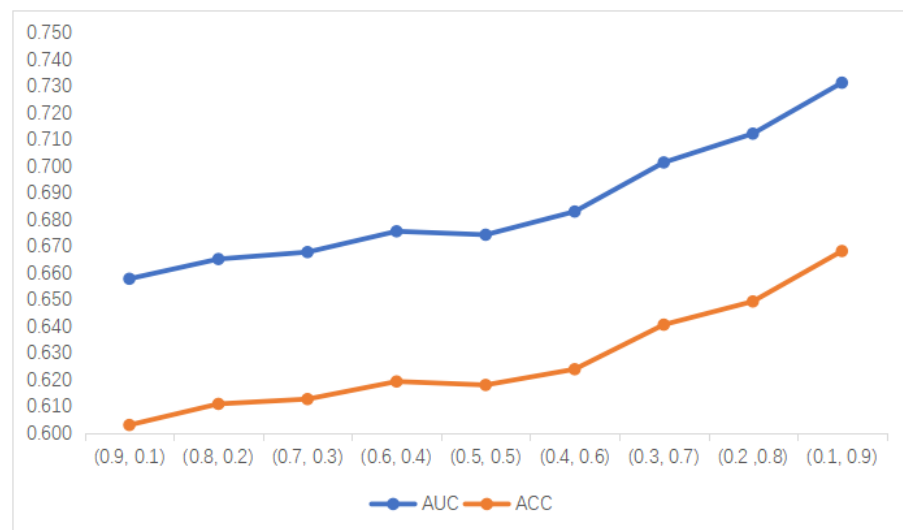


Figure 6. Impact of different hyperparameters on the performance of KGG model on Book-Crossing.

4.4.3. Analysis of Model Validity (RQ3)

To explore whether the local interest feature module and global interest feature module in the KGG model have an effect on the final performance, we removed the global interest feature module and only used the local interest feature module for training. The results are shown in Figure 7, where the orange bar chart represents the local interest feature module (KGCN model), and the blue bar chart represents the KGG model. In order to make a fair comparison, all training parameters were kept identical under the same conditions.

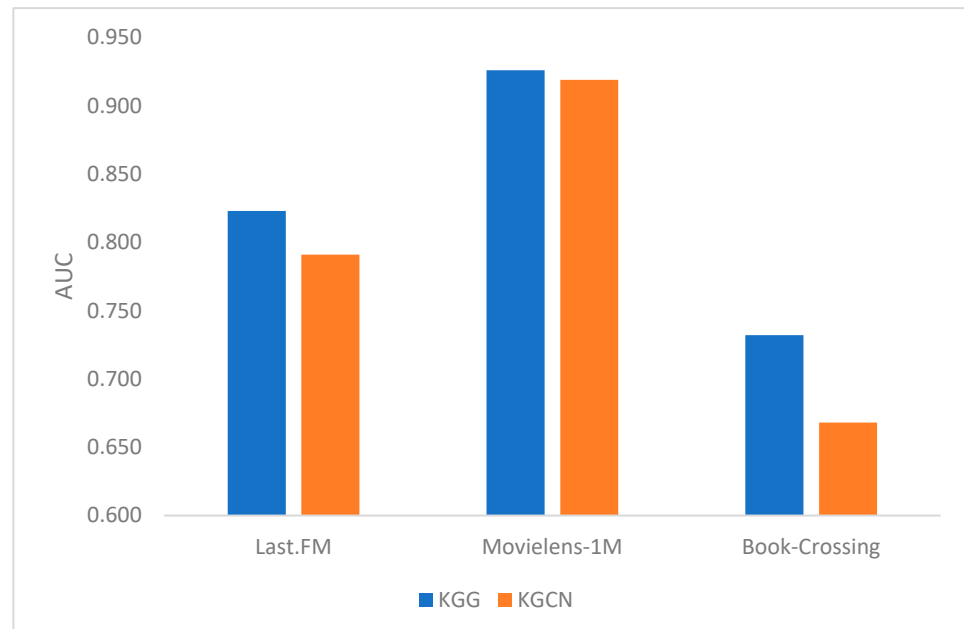


Figure 7. Results of Deletion Experiment.

The performance of the KGG and Local Interest Feature Module on Last.FM, Movielens-1M and Book-Crossing datasets are shown in Figure 7. On all three datasets, the performance of KGG outperforms that of the Local Interest Feature Module. This, to some extent, proves that using only a local interest feature module cannot sufficiently and effectively explore the user's interest distribution. The combination of local and global can not only alleviate the data sparsity problem by using different sources of information but also com-

prehensively integrate users' multiple interests to reasonably recommend items that they are more interested in.

4.4.4. Analysis of Linear Fusion Module

In order to verify the effectiveness of the linear fusion module designed in this paper, we compared three fusion methods: additive, mean, and linear weighted. From Table 3, it can be seen that the performance of using additive and mean methods are inferior to using linear weighted fusion, indicating that the proportion of local and global interests in the overall interest distribution of users is not the same, and using parameter control is more conducive to improving the performance of the model.

Table 3. Different fusion method results.

Model	Additive		Mean		Linear Weighted	
	AUC	ACC	AUC	ACC	AUC	ACC
Last.FM	0.816	0.733	0.817	0.730	0.823	0.751
Movielens-1M	0.918	0.844	0.918	0.841	0.926	0.852
Book-Crossing	0.698	0.640	0.674	0.618	0.731	0.668

4.4.5. Results in Sparse Scenarios

In KGG, we explored users' overall interest preferences using both rating information and knowledge graph information, which also further alleviates the existing data sparsity problem. To investigate the efficacy of the KGE module in sparse scenarios, we varied the ratio of the training set of MovieLens-1M from 100% to 10% (while the validation and test set are kept fixed) and reported the results of AUC in CTR prediction for all methods. The results are shown in Table 4.

Table 4. Results of partitioning training set into different ratios.

Model	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
PER	0.598	0.607	0.621	0.638	0.647	0.662	0.675	0.688	0.697	0.710
CKE	0.674	0.692	0.705	0.716	0.739	0.754	0.768	0.775	0.797	0.801
DKN	0.579	0.582	0.589	0.601	0.612	0.620	0.631	0.638	0.646	0.655
Wide&Deep	0.788	0.802	0.809	0.815	0.821	0.840	0.858	0.876	0.884	0.898
RippleNet	0.843	0.851	0.859	0.862	0.870	0.878	0.890	0.901	0.912	0.920
KGCN	0.808	0.865	0.876	0.885	0.886	0.890	0.910	0.913	0.917	0.919
Ours	0.870	0.876	0.890	0.899	0.907	0.914	0.917	0.920	0.923	0.926

In Table 4, we observed that the performance of all methods degraded with the reduction in the training set; however, the performance of KGG has been consistently lower than these comparative methods. When the training set is partitioned into 10%, the AUC score decreases by 15.8%, 15.9%, 11.6%, 12.2%, 8.4% and 12.1% for PER, CKE, DKN, Wide&Deep, RippleNet and KGCN, respectively, compared with the case when the full training set is used. However, the AUC of KGG only decreases by 6.0%, which demonstrates that KGG can still maintain a decent performance in the case of data sparsity.

5. Conclusions

This paper proposes a recommendation algorithm that combines local and global interest features. KGG utilizes Knowledge Graph Convolutional Networks to capture rich target item representations in the knowledge graph, acquires users' local interest features through matrix factorization and learns global interest features using GANs in the interaction matrix. Finally, the designed linear fusion module effectively integrates

local and global interest features. Extensive experiments on three real-world datasets demonstrate the effectiveness of the KGG model.

In the future, we plan to start from the training method of the model, from step-by-step training to joint training, to form an end-to-end model. At the same time, in the global interest feature module, we will add some unused historical interaction information (such as user comments) to improve the performance of the global interest module.

Author Contributions: Conceptualization, X.S.; methodology, X.S.; software, X.S.; validation, X.S.; formal analysis, X.S.; writing—original draft preparation, X.S.; writing—review and editing, J.Q.; supervision, Q.R.; funding acquisition, J.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science Fund for Outstanding Youth of Xinjiang Uygur Autonomous Region under Grant No.2021D01E14.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We evaluated our algorithm on three datasets. There are the download links for the three datasets. <https://grouplens.org/datasets/hetrec-2011/> (accessed on 1 January 2022); <https://grouplens.org/datasets/movielens/1m/> (accessed on 1 January 2022); <http://www2.informatik.uni-freiburg.de/~chiegler/BX/> (accessed on 1 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749. [\[CrossRef\]](#)
- Zhang, Z.-P.; Kudo, Y.; Murai, T.; Ren, Y.-G. Enhancing Recommendation Accuracy of Item-Based Collaborative Filtering via Item-Variance Weighting. *Appl. Sci.* **2019**, *9*, 1928. [\[CrossRef\]](#)
- Wang, P.; Yang, J.; Zhang, J. A Strategy toward Collaborative Filter Recommended Location Service for Privacy Protection. *Sensors* **2018**, *18*, 1522. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zhu, J.; Li, K.; Peng, J.; Qi, J. Self-Supervised Graph Attention Collaborative Filtering for Recommendation. *Electronics* **2023**, *12*, 793. [\[CrossRef\]](#)
- Dietz, L.; Kotov, A.; Meij, E. Utilizing Knowledge Graphs for Text-Centric Information Retrieval. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 1387–1390.
- Huang, J.; Zhao, W.X.; Dou, H.; Wen, J.-R.; Chang, E.Y. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 505–514.
- Huang, Z.; Chen, J.; Shen, L.; Chen, X. Fraship: A Framework to Support End-User Personalization of Smart Home Services with Runtime Knowledge Graph. In Proceedings of the Companion Proceedings of the Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 987–995.
- Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [\[CrossRef\]](#)
- Yu, X.; Ren, X.; Sun, Y.; Gu, Q.; Sturt, B.; Khandelwal, U.; Norick, B.; Han, J. Personalized entity recommendation. In Proceedings of the 7th ACM International Conference on Web Search and Data Mining, New York, NY, USA, 24–28 February 2014; pp. 283–292.
- Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.-Y. Collaborative Knowledge Base Embedding for Recommender Systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 353–362.
- Wang, H.; Zhang, F.; Xie, X.; Guo, M. Dkn. In Proceedings of the 2018 World Wide Web Conference on World Wide Web—WWW '18, Lyon, France, 23–27 April 2018; pp. 1835–1844.
- Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; Guo, M. RippleNet. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 417–426.
- Wang, H.; Zhang, F.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In Proceedings of the World Wide Web Conference, Francisco, CA, USA, 13–17 May 2019; pp. 2000–2010.

15. Wang, H.; Zhang, F.; Zhang, M.; Leskovec, J.; Zhao, M.; Li, W.; Wang, Z. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 968–977.
16. Wang, H.; Zhao, M.; Xie, X.; Li, W.; Guo, M. Knowledge Graph Convolutional Networks for Recommender Systems. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3307–3313.
17. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.-S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
18. Wang, Z.; Lin, G.; Tan, H.; Chen, Q.; Liu, X. 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, Virtual Event, 25–30 July 2020; pp. 219–228.
19. Wang, X.; Huang, T.; Wang, D.; Yuan, Y.; Liu, Z.; He, X.; Chua, T.-S. Learning Intents behind Interactions with Knowledge Graph for Recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 878–887.
20. Zeng, W.; Qin, J.; Wang, X. CKEN: Collaborative Knowledge-Aware Enhanced Network for Recommender Systems. In Proceedings of the Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, 6–9 September 2022; pp. 769–784, Proceedings, Part II.
21. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [\[CrossRef\]](#)
22. Guérin, É.; Digne, J.; Galin, É.; Peytavie, A.; Wolf, C.; Benes, B.; Martinez, B. Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Trans. Graph.* **2017**, *36*, 1–13. [\[CrossRef\]](#)
23. Deldjoo, Y.; Noia, T.D.; Merra, F.A. A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–38. [\[CrossRef\]](#)
24. Wang, J.; Yu, L.; Zhang, W.; Gong, Y.; Xu, Y.; Wang, B.; Zhang, P.; Zhang, D. Irgan. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 515–524.
25. Chae, D.-K.; Kang, J.-S.; Kim, S.-W.; Lee, J.-T. Cfgan. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 137–146.
26. Tong, Y.; Luo, Y.; Zhang, Z.; Sadiq, S.; Cui, P. Collaborative Generative Adversarial Network for Recommendation Systems. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), Macao, Macao, 8–12 April 2019; pp. 161–168.
27. Zemouri, R. Semi-Supervised Adversarial Variational Autoencoder. *Mach. Learn. Knowl. Extr.* **2020**, *2*, 361–378. [\[CrossRef\]](#)
28. Lin, Y.; Xie, Z.; Xu, B.; Xu, K.; Lin, H. Info-flow enhanced GANs for recommender. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 11–15 July 2021; pp. 1703–1707.
29. Sun, Y.; Chun, S.-J.; Lee, Y. Learned Semantic Index Structure Using Knowledge Graph Embedding and Density-Based Spatial Clustering Techniques. *Appl. Sci.* **2022**, *12*, 6713. [\[CrossRef\]](#)
30. Li, T.; Wang, W.; Li, X.; Wang, T.; Zhou, X.; Huang, M. Embedding Uncertain Temporal Knowledge Graphs. *Mathematics* **2023**, *11*, 775. [\[CrossRef\]](#)
31. Yu, D.; Yang, Y.; Zhang, R.; Wu, Y. Knowledge Embedding Based Graph Convolutional Network. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 1619–1628.
32. Yu, J.; Cai, Y.; Sun, M.; Li, P. SpaceE: Knowledge Graph Embedding by Relational Linear Transformation in the Entity Space. In Proceedings of the 33rd ACM Conference on Hypertext and Social Media, Barcelona, Spain, 28 June–1 July 2022; pp. 64–72.
33. Sha, X.; Sun, Z.; Zhang, J. Hierarchical attentive knowledge graph embedding for personalized recommendation. *Electron. Commer. Res. Appl.* **2021**, *48*, 101071. [\[CrossRef\]](#)
34. Pietrasik, M.; Reformat, M.Z. Probabilistic Coarsening for Knowledge Graph Embeddings. *Axioms* **2023**, *12*, 275. [\[CrossRef\]](#)
35. Ma, J.; Zhou, C.; Chen, Y.; Wang, Y.; Hu, G.; Qiao, Y. TeCre: A Novel Temporal Conflict Resolution Method Based on Temporal Knowledge Graph Embedding. *Information* **2023**, *14*, 155. [\[CrossRef\]](#)
36. Zhao, H.; Yao, Q.; Li, J.; Song, Y.; Lee, D.L. Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 635–644.
37. Hu, B.; Shi, C.; Zhao, W.X.; Yu, P.S. 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 & Data Mining, London, UK, 19–23 August 2018; pp. 1531–1540.
38. Luo, L.; Fang, Y.; Cao, X.; Zhang, X.; Zhang, W. Detecting Communities from Heterogeneous Graphs. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual Event, 1–5 November 2021; pp. 1170–1180.
39. Chen, H.; Li, Y.; Sun, X.; Xu, G.; Yin, H. Temporal Meta-path Guided Explainable Recommendation. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, 8–12 March 2021; pp. 1056–1064.
40. Sun, Z.; Yang, J.; Zhang, J.; Bozzon, A.; Huang, L.-K.; Xu, C. Recurrent knowledge graph embedding for effective recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2–7 October 2018; pp. 297–305.

41. Dong, C.; Ju, X.; Ma, Y. HRS: Hybrid Recommendation System based on Attention Mechanism and Knowledge Graph Embedding. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Melbourne, Australia, 14–17 December 2021; pp. 406–413.
42. Chen, M.; Zhang, W.; Zhu, Y.; Zhou, H.; Yuan, Z.; Xu, C.; Chen, H. Meta-Knowledge Transfer for Inductive Knowledge Graph Embedding. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 927–937.
43. Li, J.; Xu, Z.; Tang, Y.; Zhao, B.; Tian, H. Deep hybrid knowledge graph embedding for top-n recommendation. In Proceedings of the Web Information Systems and Applications: 17th International Conference, WISA 2020, Guangzhou, China, 23–25 September 2020; pp. 59–70, Proceedings 17.
44. Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 7–10.
45. Huang, J.; Ling, C.X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 299–310. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.