
Knowledge Interaction and Diffusion Augmentation for Knowledge Graph Recommendation

Deqing Zhang^{1,*} and Mideth Abisado²

¹*College of Computing and Information Technologies, National University, Manila, Philippine*

School of Smart Transportation Modern Industry, Anhui Sanlian University, Hefei, China

²*College of Computing and Information Technologies, National University, Manila, Philippine*

E-mail: zdqcoffee@126.com; mbabisado@national-u.edu.ph

**Corresponding Author*

Received 16 June 2025; Accepted 06 August 2025

Abstract

Knowledge graphs (KGs) provide rich semantic and structured knowledge to address data sparsity in traditional recommendation systems, yet efficiently leveraging KGs for better performance remains challenging. This paper proposes KIDRec, a knowledge graph recommendation framework integrating knowledge interaction and diffusion-augmented learning. This framework captures deep KG semantics via high-order entity interactions and uses a diffusion model to generate robust knowledge representations through “noise injection and denoising” on collaborative user embeddings and item embeddings. It then incorporates an adaptive contrastive learning module with hard negative sample enhancement and a dynamic temperature

Journal of Web Engineering, Vol. 24_7, 1045–1072.

doi: 10.13052/jwe1540-9589.2472

© 2025 River Publishers

coefficient. Experiments on Book-Crossing, Last.FM, and MovieLens20m show KIDRec outperforms baselines in AUC (0.782, 0.877, 0.979) and F1 (0.682, 0.796, 0.932), with relative AUC improvements of 0.031, 0.032, and 0.002, respectively, while alleviating data sparsity.

Keywords: Knowledge graph, recommendation, knowledge interaction, diffusion model.

1 Introduction

With the rapid development of artificial intelligence technology, a recommendation system, as one of its key applications, has become the core component of many online services and plays an indispensable role. In e-commerce platforms, accurate product recommendations not only increase product exposure, but also significantly increase conversion rates and sales. In news and streaming services, recommendation systems help users find potential points of interest in a vast amount of information and broaden their horizons. They also optimize learning paths and improve learning efficiency and outcomes through personalized learning resources and curriculum recommendations. Compared with the traditional network platform service mode, recommendation systems break the barrier that users can only filter information through search engines [1], actively help users discover their interest preferences, meet the user's personalized resource requirements, and improve the user experience.

1.1 Research Background

The collaborative filtering algorithm has been highly recognized by the industry for its advantages of efficiency and robustness [2], and has become one of the mainstream algorithms in recommendation systems. The collaborative filtering recommendation algorithm adopts the basic idea of “birds of a feather flock together” and finds similarities between groups through the behavior of groups, thereby making decisions and recommendations for users. However, the recommendations of this algorithm show obvious shortcomings in dealing with cold-start and sparse user-item interaction data, and the lack of certain interpretability is also one of the shortcomings of this algorithm. Therefore, fusing recommendation algorithms based on collaborative filtering with other auxiliary knowledge bases has become an effective way to improve the performance [3] of recommendation systems.

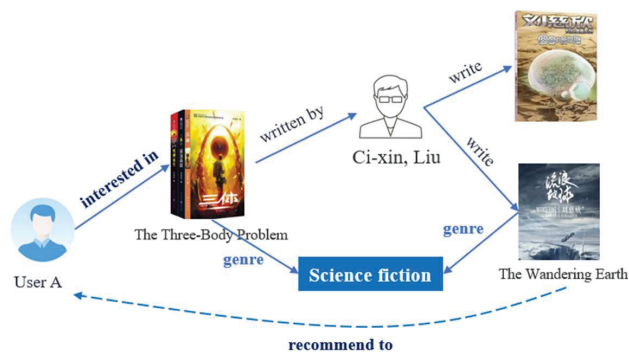


Figure 1 Example of knowledge graph recommendation.

A knowledge graph is a structured relational semantic network that stores knowledge entities and their relationships. It has excellent representation ability and usually stores their entities and relationships in the form of triples. Introducing a knowledge graph into a recommendation algorithm can not only improve the learning quality of user representation for the recommendation algorithm, but also effectively alleviates the defects of data sparsity and cold-start in the recommendation system, which has attracted extensive attention from scholars. Therefore, in recent years, recommendation algorithms based on knowledge graphs have become one of the research hotspots of recommendation systems. Not only that, but knowledge graphs can also effectively improve the accuracy, interpretability and diversity of recommendation results by virtue of their rich descriptive knowledge. As shown in Figure 1, there is a triple (UserA, interested in, “The Three-Body Problem”), (“The Three-Body Problem”, written by, Ci-xin Liu), (Ci-xin Liu, has written, “The Wandering Earth”), and as both books belong to the science fiction genre, it can be inferred that UserA may have a preference for “The Wandering Earth”. Although the knowledge graph has many advantages and abundant information, because of its high dimensionality and heterogeneity [4], how to fully and deeply mine the correlation knowledge and how to reduce the noise interference of irrelevant information in the knowledge graph are the challenges we face at present.

Currently, recommendation algorithms based on a knowledge graph can be broadly categorized into embedding-based, path-based, and propagation-based [5]. The embedding-based approach uses KGE to pre-train entities and relationships embeddings to obtain a low-dimensional vector representation of the features, and it can retain the original structural and semantic information in the knowledge graph. Its representative models include CKE [6]

and DKN [7], but the interpretability of this type of method is not strong [8], and it is more suitable for in-graph applications. Path-based methods allow users to explore potential items associated with entities through paths, and they have good interpretability. However, the method suffers from the need to manually design a meta-path or a meta-graph, which makes it difficult to optimize path selection in practice. In the case of sparse data, in particular, if the path selection is not correct, the model may overfit the training data, leading to a decrease in generalization ability. Propagation-based approaches combine embedded semantic information with paths in the knowledge graph and iteratively propagate over the entire knowledge graph to find auxiliary information for recommendation, effectively improving recommendation performance.

Propagation-based approaches have made outstanding contributions, e.g., RippleNet [9] captures high-order semantic information by iteratively propagating user preferences in the knowledge graph, which is used to construct user vector representations, improving the limitations of embedding-based and path-based recommender methods and increasing the accuracy of recommendations. KGCN [10] captures neighbor information in entities through neighborhood aggregation, which characterizes both the semantic information of the knowledge graph and the user's preference information, but the computational complexity of the model is high in the face of large-scale knowledge graph computation [11]. KGAT [12] proposes a collaborative knowledge graph (CKG) which combines the user-item graph (UIG) with the knowledge graph and supplements entity embeddings by propagating them recursively over the CKG using a graph neural network (GNN).

Although the above recommendation methods can more fully mine the semantic information and high-order connectivity of entities in the knowledge graph, they still have the problem of underutilizing collaborative information and knowledge graph information.

1.2 Research Motivation

CKAN [13] is a recommendation framework that combines collaborative filtering and a knowledge graph, encoding collaborative signals through collaborative propagation and combining them with a knowledge graph. Meanwhile, the model utilizes the attention weights to dynamically distinguish the importance of different relationship paths in the knowledge graph, which enhances the interpretability of recommendation, and models multi-hop associations between users-item-entities by stacking multi-layer

embedding propagation, which enhances the expressiveness of complex relationships.

In knowledge graph recommendation systems, the associations between users and items often present complex multi-level characteristics, and these associations are semantically transmitted through multi-hop relational paths (e.g., “user-movie-director-genre”) in the knowledge graph. Existing GCN-based models can only capture direct associations within a limited number of hops, and cannot effectively model the global implicit associations between users and items, which makes it difficult for the model to fully understand the deeper mechanisms of user preference formation.

Also, since knowledge collaborative propagation is a user-item interaction, insufficient learning of user-item representations will widen the error. In addition, the modeling of the implicit association of knowledge requires the interaction of knowledge graph triples, which can also amplify the error if the representation is not sufficiently learned. In order to minimize the problem of insufficient representation learning, contrastive learning techniques are usually introduced. However, the construction of positive samples for contrastive learning has some errors, which can lead to model performance degradation.

Contrastive learning realizes clustering semantically similar samples in the embedding space while separating irrelevant samples. However, the setting of the temperature coefficient τ significantly affects the performance of the model: too high a value of τ will lead to insufficient differentiation of negative samples in the loss function, while too low a value of τ will cause the model to pay too much attention to the hard samples and thus lead to overfitting; secondly, during the training process, most of the negative samples are easy to differentiate but do not make much contribution to the learning of the model. The real help for the model training is those hard to distinguish hard negative samples. How to effectively utilize these hard negative samples is the key to improving the performance of contrastive learning.

1.3 Contributions

In order to deal with the above challenges, we try to introduce the knowledge interaction and diffusion model into the knowledge graph recommendation in order to improve the performance of recommendation. The specific contributions are as follows:

1. We innovatively design a knowledge interaction module that realizes that the deep mining of the knowledge graph triples information

through the multi-layer explicit feature cross network and effectively captures the high-order knowledge associations implied by the user-item collaborative propagation process.

2. We innovatively use the diffusion model for knowledge graph representation learning. Through the process of “noising–denoising” of collaborative user representations and collaborative item representations, a more robust entity embedding is generated, which indirectly enhances the representation of triples and provides more discriminative input for the downstream contrastive learning module, thus achieving the effect of data enhancement and improving the performance of contrastive learning.
3. We propose an adaptive contrastive learning module that integrates hard negative sample enhancement and dynamic temperature coefficient. It aims to solve the limitations of difficult to distinguish hard negative samples and difficult to adjust temperature parameters in the existing contrastive learning, so as to make the contrastive learning have stronger distinguishing powers and achieve the goal of improving the recommendation effect.

2 Related Works

2.1 Knowledge Graph for Recommendation

Knowledge graph-based recommendation systems are favored for their ability to better understand and deal with the complex relationships between users and items.

Liu et al. [14] improved the accuracy and diversity of recommendation algorithms by incorporating the attention mechanism in knowledge graph reinforcement learning recommendation and the Random_Beam_Search search algorithm in the path inference module. Jiang et al. [15] proposed a knowledge representation learning model based on the self-attention mechanism, which learns the overall semantic information of the triple by both dynamically assigning weights to capture multi-hop information and solving the problem of insufficient use of knowledge graph semantics. Jiang et al. [16] proposed a multi-task learning scheme that enhances recommendation using knowledge graphs. The model integrates user preferences into item feature representations and optimizes neighbor aggregation through relationship importance, achieving improved recommendation performance. Zhang et al. [17] constructed an end-to-end multi-task learning framework

based on solving data sparsity and cold-start to realize joint training and feature fusion of recommender system and knowledge graph embedding through an information interaction unit. The fusion of a dynamic attention aggregation network and knowledge graph semantic information accurately models user preferences, which provides a new idea for knowledge-aware recommendation. Zhang et al. [18] applied a graph attention mechanism to a knowledge graph recommendation system to better handle the importance of different neighboring entities, and explored the potential of graph neural networks and attention mechanisms in recommendation applications.

Although knowledge graph-based recommendation shows better performance in cold-start, accuracy, and diversity, if there are triples of variable quality in the knowledge graph, the algorithm may instead be guided by the influence of noise, misleading user preference modeling and making the recommendation model less effective [19, 20].

2.2 Applications of Diffusion Models in Recommendation

Diffusion models, as a new generative paradigm, initially achieved excellent performance in the field of computer vision and NLP. In recent years, some scholars have tried to introduce the diffusion model into recommendation and have also achieved some good results.

Diffusion models in the field of recommendation systems focus on solving the core problems of data sparsity, cold-start, and noise interference [21–23]. In generic recommendation scenarios, reference [21] combines diffusion models with data augmentation paradigms to achieve robust knowledge graph representation learning, which better aligns knowledge-aware item semantics and collaborative relationship modeling. In the field of sequence recommendation, Liu et al. [24] used diffusion modeling to generate high-quality pseudo-sequence data for data augmentation in his model to effectively mitigate data sparsity and long-tail user problems. Reference [25] focuses on solving the challenges of single item representation, diverse user interests, and dynamics in sequence recommendation by using diffusion models to achieve dynamic distribution modeling, which enhances the robustness of the model by gradually adding noise and showing the injection of uncertainty. Wang et al. [26], based on the training instability of GANs and the expressive power of VAE limitations, achieved preservation of users' personalized information by reducing the noise scale during forward propagation, and used direct optimization of the reconstruction of interaction probabilities to enhance the stability of training. In addition, the diffusion

model has been extended to directions such as POI recommendation and references [27, 28] use the diffusion model in POI recommendation to alleviate the shortcomings of existing methods in new region recommendation. In the field of social recommendation, the diffusion model has also become a hot spot. References [29, 30] use a diffusion model to solve the noise pollution caused by irrelevant or false information in social relationships, RecDiff focuses on the diffusion and denoising of the hidden space, while DiffuSAR focuses more on extracting the denoised semantics from social information and knowledge transfer.

2.3 Contrastive Learning for Recommendations

Contrastive learning can effectively alleviate data sparsity and cold-start in recommendation models by self-supervised signaling with flexible data augmentation strategies and enhancing the robustness and generalization of user and item representations [31–33].

KGIC [31] integrates interaction contrastive learning by merging collaborative filtering with KG information to address the limitations of graph neural network (GNN)-based methods in sparse user-item interactions and redundant KG facts. KGCL [32] enhances robustness and reduces noise by combining KGs with contrastive learning. Wang et al. [33] employed three view enhancement strategies to develop user interaction and KG views within the MKCLR, utilizing graph contrastive learning for self-supervised learning, thereby mitigating cold-start and boosting recommendation performance. Xie et al. [34] built a user social graph and an item similarity graph from the user-item-entity graph, then achieved effective fusion of heterogeneous information through cross-graph contrastive learning, enhancing the quality of representations and alleviating data sparsity in recommendation. Sun et al. [35] leveraged semantic information in KGs through multilevel contrast learning and knowledge extraction, reducing data sparsity and decreased noise, and enhancing generalization. Xia et al. [36] suggested integrating multi-view contrastive learning with KGs and employing a multi-view generated contrastive learning filtering method to tackle noise amplification. Cui et al. [37] introduced a diffusion model based on contrastive learning, using a context-aware diffusion model to generate suitable augmentation samples, creating positive sample pairs for contrastive learning, and training the entire framework end-to-end to enhance sequential recommendation contrastive learning.

At present, the research on knowledge graph-based recommendation systems mainly covers mainstream directions such as knowledge graph

embedding, diffusion model enhancement, and contrastive learning optimization. However, existing methods require further development in mining complex high-order interactions, improving the quality of knowledge diffusion representations, and refining contrastive learning optimization strategies. This paper introduces KIDRec, an innovative recommendation framework aimed at enhancing recommendation system performance.

3 Methodology

3.1 Task Formulation

3.1.1 User-item interaction and knowledge graph

In our recommendation scenario, the input of the model consists of two major parts: user-item interaction and a knowledge graph. The specific definitions are as follows:

Suppose: a set of users $U = \{u_1, u_2, \dots, u_{N_u}\}$ and a set of items $V = \{v_1, v_2, \dots, v_{N_v}\}$ (where N_u and N_v respectively represent the total number of users and items.). The user-item interaction matrix can be defined as: $Y = \{y_{uv} | u \in U, v \in V\}$, when there is interaction between user u and item v , $y_{uv} = 1$, which means that the interaction behaviors such as clicking, watching, purchasing, rating, etc. have occurred between the user u and the item v , otherwise, $y_{uv} = 0$.

In addition, define the knowledge graph $G = \{(h, r, t) | h, t \in \varepsilon, r \in R\}$, where h and t represent the head entity and the tail entity respectively, r represents the relationship between h and t , ε denotes the set of entities, and R represents the set of relations on G .

Based on the above knowledge graph and user-item interaction matrix Y , the model will make predictions about the likelihood \hat{y}_{uv} of an interaction between user u and item v that has not interacted yet, based on the inputs from the set of users U and the set of items V , as well as from the knowledge graph G . The goal is to learn a function: $f: (U, V, Y, G) \rightarrow \mathbb{R}^{N_u \times N_v}$, such that the predicted preferences are as close as possible to the user's true preferences.

3.1.2 Diffusion model

The fundamental principle of diffusion models is rooted in the physical diffusion process, focusing on simulating data transitions from order to disorder and back to order. As illustrated in Figure 2, the model operates through two processes: forward and reverse diffusion [38].

The forward diffusion process is a process of gradually injecting Gaussian noise until it becomes pure noise. For an input sample x_0 , a time step t is first

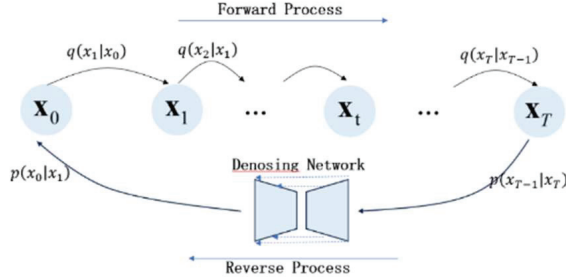


Figure 2 The two processes of the diffusion model.

randomly sampled. Subsequently, according to the preset noise schedule $\bar{\alpha}_t$, Gaussian noise ϵ of the corresponding level is injected into x_0 to obtain the noisy representation x_t :

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon \tag{1}$$

The reverse process aims to iteratively denoise the data x_T to retrieve the original data x_0 . A neural network is commonly employed to parameterize the reverse transition kernel $p_\theta(x_{t-1}|x_t, t)$. This network is typically trained to predict the noise ϵ added to x_0 , based on x_t and the time step t . The simplified training objective is:

$$\mathcal{L}_{DM} = \mathbb{E}_{t \sim [1, T], x_0 \sim q(x_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t) \right\|^2 \right] \tag{2}$$

3.2 Method

The proposed framework of knowledge-interaction and diffusion-augmentation recommendation (KIDRec) consists of a multi-order collaborative propagation layer, a knowledge diffusion module, a knowledge interaction layer, an adaptive contrastive learning module, and a prediction module (Figure 3). The framework achieves the improvement of recommendation performance through the effective fusion of each module. The multi-order collaborative propagation layer is responsible for capturing the collaborative signals of users and items, the diffusion model is used to add noise and denoise to the initial user interest and items embeddings as well as collaborative users and items embedding representations to provide a more robust representation for the model, the knowledge interaction layer then captures the deep semantic association with the knowledge graph. In order to further improve the representation quality, an adaptive contrastive learning module is designed to

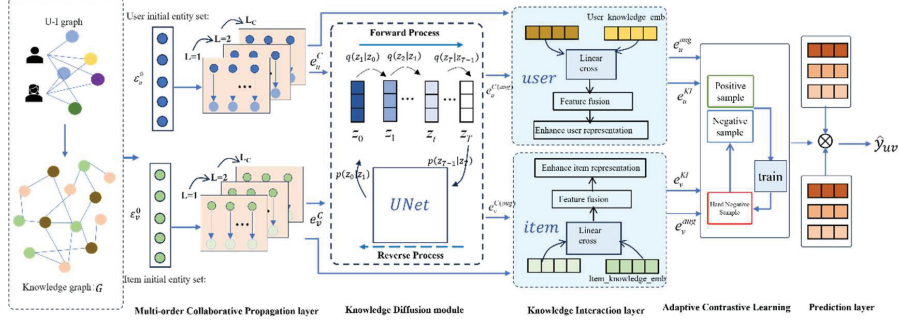


Figure 3 The proposed framework of the knowledge interaction and diffusion-augmented for knowledge graph recommendation model.

enhance the distinguishing ability of the representation through hard negative sample enhancement and dynamic temperature coefficient. Finally, the prediction layer completes the rating prediction.

3.2.1 Multi-order collaborative propagation layer

Inspired by the CKAN [13] and KGCAN [39] models, the design of the multi-order heterogeneous propagation layer of our model references the heterogeneous propagation layer of the CKAN model. This layer aims to capture users' collaborative preference information from the user behavior graph and the knowledge context information of items from the knowledge graph respectively, providing basic representations for subsequent interaction and enhancement.

1. Collaborative propagation module. Based on the premise that a user's historically interacted items can effectively reflect their interest preferences, this module constructs an enhanced user representation through the set of items the user has actually interacted with. Specifically, for user u , define their initial entity set as:

$$\varepsilon_u^0 = \{e | (v, e) \in S \text{ and } v \in \{v | y_{uv} = 1\}\} \quad (3)$$

Here, the symbol S represents the alignment set of items and entities, that is, (v, e) denotes the alignment between the item v and the entity e in the knowledge graph G .

Characterizing an item can be inferred from the users interacting with it. Users engaging with a specific item often share similar preference profiles, which can effectively illuminate and enhance the item's characterization. This

leads to the definition of the collaborative itemset for item v :

$$V_v = \{v_u | u \in \{u | y_{uv} = 1\} \text{ and } y_{uv_u} = 1\} \quad (4)$$

Further synthesis leads to an initial entity set for item v which can be expressed as:

$$\varepsilon_v^0 = \{e | (v_u, e) \in S \text{ and } v_u \in V_v\} \quad (5)$$

The above constructs the initial entity set by explicitly encoding the direct user-item interaction information, which enhances the representation while preserving the original item associations and suppresses the high-order propagation noise bias.

2. Knowledge graph propagation module. This module adopts a GCN-like architecture and automatically captures the high-order association information of entities in the knowledge graph by stacking multiple layers. The module is able to learn neighbor representations beyond multi-hops, modeling the potential relationship between users and items more comprehensively, effectively enhancing the recommender system's ability to understand user preferences. The entity sets of user u and item v are recursively defined as follows:

$$\varepsilon_u^l = \{t | (h, r, t) \in G \text{ and } h \in \varepsilon_u^{l-1}\} \quad (6)$$

$$\varepsilon_v^l = \{t | (h, r, t) \in G \text{ and } h \in \varepsilon_v^{l-1}\} \quad (7)$$

Further, the set of triples corresponding to it can be represented as:

$$triples_u^l = \{(h, r, t) | (h, r, t) \in G \text{ and } h \in \varepsilon_u^{l-1}\} \quad (8)$$

$$triples_v^l = \{(h, r, t) | (h, r, t) \in G \text{ and } h \in \varepsilon_v^{l-1}\} \quad (9)$$

In the above equation, l denotes the distance from the entity to the initial entity. With the rich semantic information in the knowledge graph, the potential relationship between user-item can be better mined. After l -layer propagation, the user and item embeddings of each layer are combined to obtain the final collaborative representation:

$$e_u^C = combine(e_u^0, e_u^1, e_u^2, \dots, e_u^l) \quad (10)$$

$$e_v^C = combine(e_v^0, e_v^1, e_v^2, \dots, e_v^l) \quad (11)$$

3.2.2 Knowledge interaction layer

In order to more accurately capture the complex, high-order feature interactions between user-item embedding in their KG contexts, a knowledge interaction layer is introduced to deeply integrate the collaborative information with the KG embeddings in the vector space by establishing a mechanism for bidirectional interactions between users and items.

The inputs of the knowledge interaction layer are the user collaborative representation e_u^C and the item collaborative representation e_v^C output from the multi-order collaborative propagation layer. For each triple (v, r, t) or (h, r, v) related to item v in the knowledge graph G , an interaction feature vector is constructed. The model employs a CrossNet structure to capture the high-order interactions between these features. A key advantage of CrossNet over traditional GNN message-passing mechanisms lies in its ability to explicitly model heterogeneous relational semantics in knowledge graphs. Unlike GNNs, which primarily aggregate neighborhood information through implicit weighted summation and may blur the distinct characteristics of different relationship types (e.g., symmetric “similar to” vs. asymmetric “written by”), CrossNet achieves explicit feature crossing via its layer-wise multiplicative operations. This allows it to preserve and amplify the unique patterns of heterogeneous relationships: for instance, in handling a triple (movie, directed by, director) and another (movie, belongs to, genre), CrossNet can distinguish the directional dependency in “directed by” and the categorical association in “belongs to” through separate cross terms, whereas GNNs might conflate these relational nuances during neighborhood aggregation.

Firstly, the collaborative embedding of item v in e_v^C is interacted with its related knowledge graph information. For each triple related to item v , an input vector is constructed:

$$x_{v,0} = \text{concat}(e_v^C, e_r^{(0)}, e_t^{(0)}) \quad (12)$$

Then, the vector is fed into the CrossNet of the L_{cross} layer, and each layer of CrossNet is calculated as:

$$x_{k+1} = x_{v,0} x_k^T w_k + b_k + x_k \quad (13)$$

where $k \in [0, L_{Cross} - 1]$, $x_k \in \mathbb{R}^d$ is the output of the k th layer cross network and $w_k, b_k \in \mathbb{R}^d$ are the weights and biases of the layer. After the L_{cross} layer CrossNet, the embedding representation e_v^{KI} of the item v after knowledge interaction can be obtained, and similarly, the user embedding e_u^{KI} can be obtained.

3.2.3 Knowledge diffusion module

The multi-order collaborative propagation and knowledge interaction module constructs user and item representations. However, inadequate initial representations may lead to error amplification in propagation and affect the comprehensive performance of recommendations. Inspired by the powerful generative capabilities of diffusion models [21, 26, 28], and in order to enhance the robustness and generalization of the knowledge representation, we introduce the knowledge diffusion module, which diffuses and denoises the collaborative user representation and item representation to achieve enhanced representation.

Collaborative user embedding representations and item embedding representations are chosen as diffusion units is that the embedding representations of users and items are continuous vectors, which are in line with the data types that standard diffusion models are good at processing, and can be directly processed using mature diffusion models. In addition, this method only performs diffusion and enhancement at the feature level, without changing the discrete topological structure of the graph, so the integrity of the knowledge graph structure will not be compromised. Using a diffusion model for this enables the model to learn more robust representations, enhance generalization ability, and provide a stable foundation for downstream recommendations.

Taking the item side as an example, the process of applying the diffusion model for embedding representation z_0 through “noise injection and denoising” is as follows:

1. Forward process. In the forward noise injecting process, by gradually adding Gaussian noise to z_0 in T time steps, we can get:

$$q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (14)$$

where \mathbf{z}_t is the item representation after adding noise at time t . \mathcal{N} represents Gaussian distribution, $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$, ϵ is Gaussian noise, and $\epsilon \sim (0, I)$.

2. Reverse denoising process. The reverse process aims to gradually restore the item embedding representation and user representation that have become Gaussian noise data to clear representation. We train a neural network $\epsilon_\theta(\mathbf{z}_t, t)$ to predict the noise added at time t . The process can be described by the following formula:

$$p_\theta(z_{t-1}|z_t) = \mathcal{N}\left(z_{t-1}; \mu_\theta(z_t, t), \sum_\theta(z_t, t)\right) \quad (15)$$

In this formula, $\mu_\theta(z_t, t)$ and $\Sigma_\theta(z_t, t)$ represent the conditional mean and covariance learned by the neural network with parameter θ .

The training loss is:

$$\mathcal{L}_{diff} = \mathbb{E}_{t, \mathbf{z}_0, \epsilon} [\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t)\|^2] \quad (16)$$

After training, the diffusion model generates smoother augmented user embedding representations and item embedding representations. This enhanced representation $\widehat{\mathbf{z}}_0$ will aiming to further improve the enhanced item representation $e_v^{KI(aug)}$ (abbreviated as e_v^{aug}) and user representation $e_u^{KI(aug)}$ (abbreviated as e_u^{aug}) after knowledge interaction, making them more robust. They will serve as the input basis for subsequent contrastive learning tasks.

3.2.4 Adaptive contrastive learning module

In order to further improve the quality of representations and guide the model to learn more discriminative semantic information, we designed an adaptive contrastive learning module. It aims to bring semantically similar samples closer together in the embedding space while pushing semantically dissimilar samples away. We have employed hard negative mining and dynamic temperature adjustment strategies to optimize the embedding structure. Specifically, the knowledge diffusion module (after noise addition and denoising) refines the entity embeddings of collaborative users and items and enhances the robustness of their representations, ensuring that the user and item representations fed into contrastive learning retain core semantic relationships while reducing noise interference. This enables more accurate similarity calculations between positive and negative samples, laying a reliable foundation for hard negative sample mining and dynamic temperature adjustment.

It is known that e_u^{aug} and e_v^{aug} represent the embedding representations of users and items enhanced by the knowledge diffusion module and then an InfoNCE contrastive loss function with the ‘‘hard negative sample mining’’ strategy is adopted to optimize the model. This module will conduct contrastive learning separately for both of them and their original embedding representations. For each entity’s original embedding and the enhanced embedding of the diffusion model, a positive sample pair is constructed.

1. Hard negative sample enhancement. In contrastive learning, many negative samples are too dissimilar to the anchor samples, resulting in minimal contribution to the gradient. Thus, simply labeling all other samples within a batch as negative leads to inefficiency. To address the challenge

of distinguishing between positive and negative samples, a mechanism for mining hard negative samples has been introduced.

Taking the user side as an example, a positive sample pair is constructed for the original representation $e_{u_i}^{KI}$ and enhanced representation $e_{u_i}^{aug}$ of each user entity u_i to calculate their similarity $s_{pos} = sim(e_{u_i}^{KI}, e_{u_i}^{aug})$. Other entities are used as potential negative samples, for each negative sample user $u_j (j \neq i)$, calculate its similarity $s_{neg,j} = sim(e_{u_i}^{KI}, e_{u_j}^{KI})$ with the anchor user u_i . If the similarity $s_{neg,j}$ between negative samples and user u_i is relatively high, i.e. $s_{neg,j} > s_{pos} - \delta_{margin}$ (δ_{margin} is a boundary hyperparameter), then it is considered to be a hard negative sample for u_i , and contrastive learning only calculates the contrastive loss for these selected hard negative sample sets.

2. Adaptive temperature coefficient. The temperature coefficient τ affects how strongly the model penalizes negative samples in contrastive learning. Our tuning strategy is: as the training epoch increases, the temperature parameter τ gradually decreases. At the beginning of training, a higher temperature allows the model to explore a wider sample space; as training proceeds, gradually lowering the temperature allows the model to pay more attention to hard negative samples, thus improving the model's discriminative ability. As shown in Equation (17), let τ_{adapt} decrease gradually with the increase of training epoch:

$$\tau_{adapt}^{(epoch)} = \max \left(\tau_{min}, \tau_{initial} \cdot \left(1 - \frac{epoch}{max_epochs} \right) \right) \quad (17)$$

where $\tau_{initial}$ is the initial temperature, τ_{min} is the lower temperature limit, $epoch$ is the current training epoch, and $(1 - \frac{epoch}{max_epochs})$ is the core for achieving attenuation.

Combining hard negative sample enhancement and dynamic temperature coefficient, the contrastive learning loss $\mathcal{L}_{CL}^{(u)}$ for user u is defined as:

$$\begin{aligned} \mathcal{L}_{CL}^{(u)} = & -\log \frac{\exp(sim(e_{u_i}^{KI}, e_{u_i}^{aug})/\tau_{adapt})}{\exp(sim(e_{u_i}^{KI}, e_{u_i}^{aug})/\tau_{adapt})} \\ & + \sum_{u_j \in H_{u_i,hard}^-} \exp(sim(e_{u_i}^{KI}, e_{u_j}^{KI})/\tau_{adapt}) \end{aligned} \quad (18)$$

where $sim(\cdot)$ is the similarity function. User side contrastive learning loss takes the mean of all entity contrastive loss. Similarly, the loss on the item side can be calculated. The total comparative loss is the sum of the two, that is, $\mathcal{L}_{CL} = \mathcal{L}_{CL}^{(u)} + \mathcal{L}_{CL}^{(v)}$.

3.2.5 Prediction layer

Through the previous steps, the model calculates the final user representation e_u^{KI} and item representation e_v^{KI} , and the prediction layer calculates the user u 's preference score \hat{y}_{uv} for item v by embedding aggregation with scoring calculation. The method used here is to calculate the inner product to obtain the preliminary rating:

$$score_{uv} = (e_u^{KI})^\top e_v^{KI} \quad (19)$$

Then further apply the Sigmoid activation function $\sigma(\cdot)$ to obtain the preference score \hat{y}_{uv} of user u for item v :

$$\hat{y}_{uv} = \sigma(score_{uv}) = \frac{1}{1 + \exp(-score_{uv})} \quad (20)$$

The main recommendation task was optimized using binary cross-entropy loss:

$$\mathcal{L}_{Rec} = -\frac{1}{N} \sum_{(u,v) \in \mathcal{O}^+ \cup \mathcal{O}^-} (y_{uv} \log(\hat{y}_{uv}) + (1 - y_{uv}) \log(1 - \hat{y}_{uv})) \quad (21)$$

where \mathcal{O}^+ is the set of positive samples and \mathcal{O}^- is the set of negative samples and N represents the sample size. Ultimately, the total loss function of the model is a weighted sum of the recommendation loss, knowledge diffusion loss, and contrastive learning loss, while an L2 regularization term is added to prevent overfitting from occurring.

$$\mathcal{L}_{total} = \mathcal{L}_{Rec} + \lambda_{diff} \mathcal{L}_{diff} + \lambda_{CL} \mathcal{L}_{CL} \quad (22)$$

\mathcal{L}_{total} denotes the final total objective loss function, and λ_{diff} and λ_{CL} denote the weight parameters of the diffusion module and adaptive contrastive learning module, respectively.

4 Experiment and Results

In this section, we will present the experimental results and systematically evaluate the performance of the model.

4.1 Experiment Settings

4.1.1 Datasets

To facilitate the evaluation and comparison of model performance, three public datasets commonly used in the research field of recommendation

Table 1 Statistical information of three experimental datasets

Information	Book-Crossing	Last.FM	Movielens20m
Num of users	17860	1872	138159
Num of items	14967	3846	16954
Num of entities	77903	9366	102569
Num of relations	25	60	32
Num of triples	151500	15518	499474

algorithms: Book-Crossing [40], Last.FM [41], and MovieLens20m [42] were selected as the subjects of the experiment. The knowledge graphs of the datasets we used comes from <https://github.com/weberrr/CKAN> publicly available data. The statistical information of the dataset is shown in Table 1.

4.1.2 Experimental environment and parameter settings

The environment for this experiment is as follows: PyTorch, CUDA version 12.1. The Adam optimizer was used, and the Xavier initializer [43] was selected. This choice is motivated by the need to align with the characteristics of knowledge graphs, which typically exhibit entity type diversity and varying degrees of relationship sparsity.

Knowledge graphs in recommendation scenarios consist of heterogeneous entities (e.g., users, items, Book-Authors in Book-Crossing; artists, albums, tags in Last.FM) with uneven connectivity – some entities participate in numerous triples (e.g., popular movies in MovieLens20m), while others have sparse relational links. Xavier initialization addresses this by scaling initial weights based on the number of input and output neurons, maintaining consistent variance across layers. This balance is critical; for densely connected entities, it prevents overly large initial embeddings that might cause early saturation; for sparsely connected entities, it avoids excessively small values that could impede gradient propagation during training.

Notably, while entity-specific priors (e.g., popularity metrics or relationship frequencies) might seem relevant for initialization, Xavier’s unbiased approach provides a neutral starting point. This is advantageous because subsequent modules – particularly the knowledge diffusion module and multi-order propagation layer – are designed to refine embeddings using the knowledge graph’s intrinsic structural information. A neutral initialization ensures the model learns representations driven by semantic relationships rather than pre-existing biases toward popular entities.

Table 2 Parameter settings

Parameter	Value
batch_size	4096
learning_rate	5×10^{-4} , 2×10^{-3} , 3.5×10^{-3}
n_layer	3
dimension	8,16,32,64,128
l2_weight	1e-4

Table 3 Results of the AUC and F1 values for the CTR experiment

Model	Book-Crossing		Last.FM		Movielens20m	
	AUC	F1	AUC	F1	AUC	F1
CKE	0.673	0.619	0.744	0.673	0.925	0.870
RippleNet	0.721	0.646	0.789	0.718	0.976	0.929
KGCN	0.692	0.639	0.803	0.714	<u>0.977</u>	<u>0.930</u>
KGAT	0.729	0.653	0.830	0.744	0.975	0.926
CKAN	<u>0.751</u>	<u>0.671</u>	<u>0.845</u>	<u>0.771</u>	0.976	0.921
KIDRec	0.782	0.682	0.877	0.796	0.979	0.932
Improve	0.031	0.011	0.032	0.025	0.002	0.002

In this experiment, the training set, test set, and validation set of each dataset were divided at a ratio of 6:2:2. The parameter settings are shown in Table 2.

4.1.3 Evaluation indicators

To comprehensively evaluate the performance of the model in predicting user interaction probability, we constructed a CTR task and adopted the AUC and F1 as evaluation metrics for the model. The AUC is an important metric for measuring the overall ranking ability of the model for positive and negative samples. The closer the AUC is to 1, the stronger the model's discrimination ability. The latter is the harmonic mean of precision and recall when the model identifies positive samples. The larger the value, the better the model performs in terms of precision and recall.

4.2 Experiment Results

As can be seen from the data in Table 3, the AUC results of our model on the three datasets of Book-Crossing, Last.FM, and Movielens20m are all superior to those of the advanced baseline models. The innovation of the KIDRec model lies in the knowledge interaction, which realizes multi-layer explicit interaction of the triples in the KG. The diffusion model is used to

denoise the original collaborative user and item embeddings representation data to achieve the data augmentation effect of contrastive learning, and hard negative sample mining is used to further improve the performance of contrastive learning. Through vertical comparison with other baselines, it can be found that the AUC values of the KIDRec model on the three datasets all achieve the best results, with relative improvements of 0.031, 0.032, and 0.002 compared with the highest scores of the baselines respectively. From the horizontal comparison of KIDRec on the three datasets, the best effect is achieved on MovieLens20m, which is partly attributed to the richer and more comprehensive knowledge information contained within MovieLens20m.

4.3 Analysis of Parameter Influence

4.3.1 Analysis of the influence of batch_size on the AUC results

To verify the influence of batch_size on performance, we conducted four groups of experiments with values of (512, 1024, 2048, 4096). As shown in Figure 4, the AUC results of tests on three datasets all show an upward trend with the increase of batch_size, indicating that increasing batch_size can provide more stable gradient estimation and has a positive impact on the model performance. We also found that, compared with the Book-Crossing and Last.FM datasets, the change in MovieLens20m is relatively gentle. A possible reason is that the knowledge graph data of the latter is more saturated, enabling the model to learn the data distribution well even with a small batch_size. The former data is sparser, and increasing batch_size can effectively aggregate information, alleviating data sparsity and potentially enhancing the model's ability to capture long-tail relationships. However, continuously increasing batch_size may lead to slower convergence, increased memory usage, and decreased efficiency. It may also cause the model to rely too much on the training data and reduce its generalization

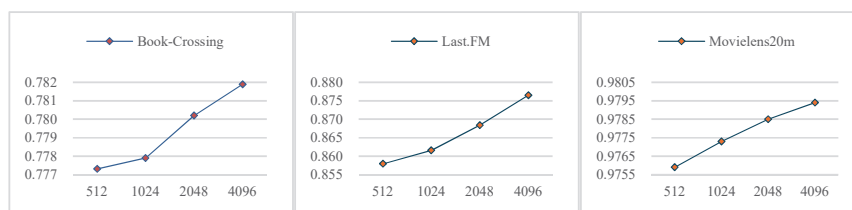


Figure 4 The AUC performance of different batch_size on three datasets.

Table 4 The AUC performance of different depth of layer on three datasets

Depth_of_layer	Book-Crossing	Last.FM	Movielens20m
1	0.7730	0.8688	0.9793
2	0.7736	0.8716	<u>0.9794</u>
3	<u>0.7819</u>	<u>0.8765</u>	<u>0.9794</u>

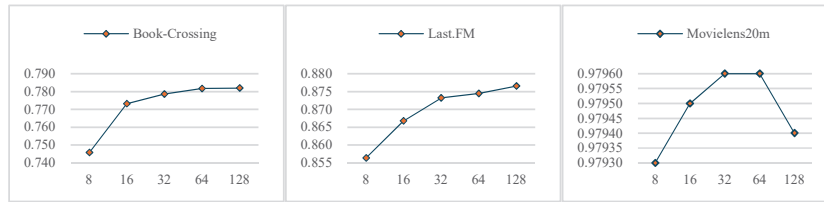


Figure 5 The AUC performance of different dimensions on three datasets.

ability. Therefore, in this experiment, we set 4096 as the batch_size for training.

4.3.2 Analysis of the influence of depth of layer on the AUC results

As shown in Table 4, we tested the performance of different depth_of_layer on three datasets respectively. From the overall AUC performance, MovieLens20m performs optimally, while Book-Crossing shows weaker performance, indicating that the influence of depth_of_layer on the recommendation effect is associated with the sparsity of the datasets to some extent. Both Book-Crossing and Last.FM achieve the best AUC results when depth_of_layer is set to 3, while the performance of MovieLens20m remains basically unchanged. A possible reason is that, when data is sparse, shallow layers cannot capture complex relationships, and high-order connection propagation can better capture user preferences. Since MovieLens20m itself has a large amount of rich information and has achieved good performance, the change of depth_of_layer has a weak influence on this dataset.

4.3.3 Analysis of the influence of dimension of embeddings on the AUC results

We also tested the impact of changes in the dimension (d) of embeddings on the AUC results. Figure 5 more intuitively presents the changes in the impact of the dimension data in Table 5 on the AUC performance. As shown in Figure 5, d exhibits different differences on different datasets. For the Book-Crossing and Last.FM datasets, when $d \leq 32$, the performance of the model

Table 5 The AUC performance of different dimensions d of embeddings on three datasets

Dimension	Book-Crossing	Last.FM	Movielens20m
8	0.7458	0.8564	0.9793
16	0.7731	0.8668	0.9795
32	0.7785	0.8732	0.9796
64	0.7816	0.8744	0.9796
128	0.7819	0.8765	0.9794

Table 6 The results of the ablation experiment

	Book-Crossing		Last.FM		Movielens20m	
	AUC	F1	AUC	F1	AUC	F1
w/o KD	0.7812	0.6798	0.8708	0.7798	0.9794	0.9316
w/o KI	0.6885	0.6433	0.8332	0.7667	0.9601	0.9089
w/o ACL	0.7800	0.6787	0.8718	0.7773	0.9792	0.9311
KIDRec	0.7819	0.6818	0.8765	0.7957	0.9794	0.9318

shows a rapid upward trend, indicating that smaller embedding dimensions may not be able to fully capture the complex features of users and items, limiting the expressive power of the model. As d increases, the AUC tends to flatten out or may show a downward trend, which means that excessively high d may also increase the risk of overfitting. For MovieLens20m, it already contains rich knowledge information and has shown good performance in the model. The change in d has only a slight impact on the performance of the AUC; when d is too large, it is prone to cause overfitting.

4.4 Ablation Experiment

To thoroughly investigate the contributions of the three key components of the KIDRec model, we conducted ablation experiments. Table 6 records the AUC and F1 values of the KIDRec model with the removal of the knowledge diffusion module, knowledge interaction layer, and adaptive contrastive learning module on three datasets.

The ablation experiment results on three datasets demonstrate the contribution of each module to the overall performance of the model. Removing any module may result in a decrease in the model’s AUC and F1 metrics, indicating the necessity of designing each module.

1. The knowledge interaction layer is more sensitive to the impact on the AUC performance of the model. When this module is removed, the AUC values on all three datasets decline significantly. Specifically, the AUC

performance on the Book-Crossing dataset drops by 0.0934, and those on the Last.FM and Movielens20m datasets also decrease by 0.0433 and 0.0193 respectively. This result fully demonstrates the effectiveness of explicitly modeling the high-order interactions among entities in the knowledge graph for capturing the deep semantics of items, which is more effective in the case of data sparsity and shows its potential in alleviating data sparsity.

2. The knowledge diffusion module and adaptive contrastive learning module have also made positive contributions to the model. When these two modules are removed separately, the performance of the AUC and F1 also decreases. The results indicate that the use of diffusion models can denoise and enhance the data representation of collaborative user interest representation and item representation, enabling KG representation to better support downstream recommendations. When the adaptive contrastive learning module is removed, there is a different decrease on different datasets, which proves that enhancing representation discrimination through hard negative sample mining has improved recommendation performance. Another possible reason is that some highly discriminative representations have already been learned in the upstream components of the model, so the ACL module's overall performance improvement effect on the model is not very prominent.

5 Conclusions

In the design work of the knowledge graph recommendation framework based on knowledge interaction and diffusion model enhancement, we innovatively introduced the knowledge interaction layer, diffusion model, and adaptive contrastive learning module to make the model achieve better AUC performance. The knowledge of the KG triples is explicitly interacted with in multiple layers to achieve the explicit association between users and items in the propagation process. Through the powerful generation function of the diffusion model, the collaborative user interest representation and item representation are diffused and denoised to achieve indirect enhancement of the knowledge graph data. The hard negative sample enhancement and dynamic temperature coefficient design in the adaptive contrastive learning module further enhance the distinguishing ability of the representation, achieving a significant improvement in the model recommendation performance. In this work, we have fully verified the effectiveness of the model on three commonly used data sets through a large number of experiments.

In the future, we will also try solutions to achieve how to reduce the complexity of the model while ensuring performance.

Acknowledgments

Special thanks for the learning platform provided by the National University and Anhui Sanlian University. The work was supported by “Outstanding Young Teacher Training Program of Anhui Province” (Grant No. YQYB2023079), and “Project of Quality Engineering of Anhui Province” (Grant No. 2023sx141), and “Project of Natural Science of Anhui Province” (Grant No. 2024AH050498).

References

- [1] Y. Guo, “Research on Recommender Method Based on Knowledge Graph and User Preferences,” Dalian Maritime University, Dalian, China, 2023.
- [2] J. Gu, R. Wang, N. Li and S. Zhang, “Knowledge graph attention network fusing collaborative filtering information,” *Journal of Computer Applications*, vol. 42, no. 4, pp. 1087–1092, 2022.
- [3] D. Lu, D. Zhu and H. Du, et al., “Fusion Recommendation System Based on Collaborative Filtering and Knowledge Graph,” *Computer Systems Science & Engineering*, vol. 42, no. 3, pp. 1133–1146, 2022.
- [4] H. Liu, X. Li and L. Hu, et al., “Knowledge graph driven recommendation model of graph neural network,” *Journal of Computer Applications*, vol. 41, no. 7, pp. 1864–1870, 2021.
- [5] Q. Guo, F. Zhang, C. Qin, H. Zhu, X. Xie, H. Xiong and Q. He, “A Survey on Knowledge Graph-Based Recommender Systems,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3549–3568, 2020.
- [6] F. Zhang, N. J. Yuan, D. Lian, X. Xie and W. Ma, “Collaborative knowledge base embedding for recommender systems,” In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’16)*, pp. 353–362, 2016.
- [7] H. Wang, F. Zhang, X. Xie and M. Guo, “DKN: deep knowledge-aware network for news recommendation,” In *Proc. 2018 World Wide Web Conference (WWW’18)*, pp. 1835–1844, 2018.

- [8] Y. Zhao, L. Liu, H. Wang, H. Han and D. Pei, "A Survey of Knowledge Graph Recommender System Research," *Journal of Frontiers of Computer Science and Technology*, vol. 17, no. 4, pp. 771–791, 2023.
- [9] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems," in *Proc. 27th ACM International Conference on Information and Knowledge Management (CIKM'18)*, pp. 417–426, 2018.
- [10] H. Wang, M. Zhao, X. Xie, W. Li and M. Guo, "Knowledge Graph Convolutional Networks for Recommender Systems," in *Proc. 2019 World Wide Web Conference (WWW'19)*, pp. 3307–3313, 2019.
- [11] H. Ma and P. Hu, "Knowledge Graph-based Recommendation Model with Bipartite Knowledge Aware GCN," *Computer Systems & Applications*, vol. 33, no. 1, pp. 289–296, 2024.
- [12] X. Wang, X. He, Y. Cao, M. Liu and T. Chua, "Knowledge Graph Attention Network for Recommendation," in *Proc. Knowledge Discovery & Data Mining (KDD'19)*, pp. 950–958, 2019.
- [13] Z. Wang, G. Lin, H. Tan, Q. Chen and X. Liu, "Collaborative Knowledge-aware Attentive Network for Recommender Systems," in *Proc. SIGIR'20*, pp. 219–228, 2020.
- [14] J. Liu, "Research on Reinforcement Learning Recommendation Algorithm Based on Knowledge Graph," *Henan Science and Technology*, no. 20, pp. 31–36, 2024.
- [15] Z. Jiang, C. Chi and Y. Zhan., "Let Knowledge Make Recommendations for You," *IEEE Access*, vol. 9, pp. 118194–118204, 2021.
- [16] M. Jiang, M. Li, W. Cao, M. Yang and L. Zhou, "Multi-task convolutional deep neural network for recommendation based on knowledge graphs," *Neurocomputing*, vol. 619, pp. 129136, pp. 1–13, 2025.
- [17] D. Zhang, X. Yang, L. Liu and Q. Liu, "A Knowledge Graph-Enhanced Attention Aggregation Network for Making Recommendations," *Applied sciences*, vol. 11, no. 21, pp. 10432:1–12, 2021.
- [18] X. Zhang, Q. Deng and X. Liu, "Knowledge Graph Recommendation Algorithm Combined with Graph Attention Mechanism," *Computer Science*, vol. 50, pp. 464–470, 2023.
- [19] S. Zhu, Y. Zhang, L. Zhang, X. Chen and L. Zhao, "MAGIC: Noise Mitigation and Knowledge Alignment for Knowledge Graph-Based Multi-modal Recommendation," in *Proc. 2025 International Conference on Multimedia Retrieval (ICMR'25)*, pp. 1931–1939, 2025.

- [20] M. Cui, S. Wu, H. Chen and X. Zhang, “RsDiff: Rational score based knowledge graph diffusion for recommendation,” Vol. 717, 122292, 2025. <https://doi.org/10.1016/j.ins.2025.122292>.
- [21] Y. Jiang, Y. Yang, L. Xia and C. Huang. “DiffKG: Knowledge Graph Diffusion Model for Recommendation,” In Proc. 17th ACM International Conference on Web Search and Data Mining, (WSDM’24), pp. 313–321, 2024.
- [22] Y. Jiang, L. Xia, W. Wei, D.Luo, K. Lin. C. Huang, “DiffMM: Multi-Modal Diffusion Model for Recommendation,” Proceedings of the 32nd ACM International Conference on Multimedia (MM’24:), pp. 7591–7599, 2024.
- [23] H. Dong, H. Liang, J. Yu and K. Gai, “DICES: Diffusion-Based Contrastive Learning with Knowledge Graphs for Recommendation,” In Proc. Knowledge Science, Engineering and Management(KSEM), pp. 117–129, 2024.
- [24] Q. Liu, F. Yan and X. Zhao, et al., “Diffusion Augmentation for Sequential Recommendation,” In Proc. International Conference on Information and Knowledge Management(CIKM ’23), pp. 1576–1586, 2023.
- [25] Z. Li, C. Li and A. Sun, “DiffuRec: A Diffusion Model for Sequential Recommendation,” ACM Transactions on Information Systems, vol. 37, no. 4, pp. 111:1–28, 2023.
- [26] W. Wang, Y. Xu, F. Feng, X. Lin, X. He and T. Chua, “Diffusion Recommender Model,” In Proc. SIGIR’23, pp. 832–841, 2023.
- [27] J. Zuo and Y. Zhang, “Diff-DGMN: A Diffusion-Based Dual Graph Multiattention Network for POI Recommendation,” IEEE Internet of Things Journal, vol. 11, no. 23, pp. 38393–38409, 2024.
- [28] Y. Qin, H. Wu, W Ju, X. Luo and M. Zhang, “A Diffusion Model for POI Recommendation,” Tansactions on Information Systems, vol. 42, no. 2, pp. 54:1–27, 2023.
- [29] Z. Li,L. Xia and C. Huang, “RecDiff:Diffusion Model for Social Recommendation,” In Proc. CIKM ’24, pp. 1346–1355, 2024.
- [30] X. Zang, H. Xia and Y. Liu, “Diffusion social augmentation for social recommendation,” The Journal of Supercomputing, vol. 81, pp. 208:1–28, 2025.
- [31] D. Zou, W. Wei and Z. Wang, “Improving Knowledge-aware Recommendation with Multi-level Interactive Contrastive Learning,” Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM’22), pp. 2817–2826, 2022.

- [32] Y. Yang, C. Huang, L. Xia and C. Li, “Knowledge Graph Contrastive Learning for Recommendation,” In Proc. SIGIR’22, pp. 1434–1443, 2022.
- [33] G. Wang and H. Jiang, “Recommendation Algorithm Incorporating Multi-view Contrastive Learning and Knowledge Graph,” *Computer Systems & Applications*, vol. 34, no. 6, pp. 118–127, 2025.
- [34] X. Xie, Z. Xie, Y. Liu, J. Wang and Q. Zhan, “Knowledge-Aware Multi-view Contrastive Learning for Recommendation,” *Expert Systems with Applications*, vol. 57, pp. 36:1–25, 2025.
- [35] T. Sun, X. Zhang, Y. Chen, H. Zou and Q. Wu, “A Multi-Level Contrastive Learning Framework for Knowledge Graph-Based Recommendation Systems,” In Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC’2024), pp. 456–462, 2024.
- [36] B. Xia, J. Qin, L. Han, A. Gao and C. Ma, “Knowledge filter contrastive learning for recommendation,” *Knowledge Information System*, vol. 66, pp. 6697–6716, 2024.
- [37] Z. Cui, H. Wu, B. He, J. Cheng and C. Ma, “Context Matters: Enhancing Sequential Recommendation with Context-aware Diffusion-based Contrastive Learning,” In Proc. the 33rd ACM International Conference on Information and Knowledge Management (CIKM’24), pp. 404–414, 2024.
- [38] Jonathan Ho, Ajay Jain and Pieter Abbeel, “Denoising Diffusion Probabilistic Models,” In Proc. 34th Conference on Neural Information Processing Systems (NeurIPS’2020), vol. 33, pp. 6840–6851, 2020.
- [39] E. Elahi, S. Anwar, B. Shah, Z. Halim., A. Ullah, I. Rida and M. Waqas, “Knowledge Graph Enhanced Contextualized Attention-Based Network for Responsible User-Specific Recommendation,” *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 4, pp. 83:1–24, 2024.
- [40] C.-N. Ziegler, S. M. McNee, J. A. Konstan and G. Lausen. “Improving Recommendation Lists Through Topic Diversification,” In Proc. 14th international conference on World Wide Web, pp. 22–32, 2005.
- [41] I. Cantador, P. Brusilovsky, and T. Kuflik, “2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011),” in Proc. 5th ACM Conf. Recomm. Syst., pp. 387–388, 2011.
- [42] F. Maxwell Harper and Joseph A. Konstan, “The MovieLens Datasets: History and Context,” *ACM Transactions on Interactive Intelligent Systems (TiS)*, vol. 5, no. 4, pp. 19:1–19, 2015.
- [43] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” In AISTATS. pp. 249–256, 2010.

Biographies



Deqing Zhang received her bachelor's degree in Computer Science and Technology in 2006 and her master's degree in Management in 2009. Currently, she is a teacher of Computer Science and Technology at Anhui Sanlian University. Her main research fields include recommendation system research and deep learning.



Mideth Abisado received her Master of Science degree in computer science in 2016 and her Doctorate degree in information technology from the Technological Institute of the Philippines in 2019. She is a Professor at the College of Computing and Information Technologies, National University. She has worked in artificial intelligence research, with a particular interest in affective computing and natural language processing, and she has published many research papers.