Knowledge Graph-based Recommendation via Generative Denoising and Multi-Level Contrastive Learning

Jibing Gong[1,2], Xinchao Feng[1], Qian Zang[1], Yi Zhao[2*], Le Chen[1], Yu Qiu[3], Jun Feng[4], Haomin Song[4], Liangying Peng[4]

[1]School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
[2]Research Institute of Yanshan University, Shenzhen Virtual University Park, Shenzhen 518063, China
[3]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[4]Zhipu AI, Beijing 100085, China

Keywords: Knowledge graph; Denoising; Contrastive learning

Citation: Gong J.B., Feng X.C., Zang Q., et al.: Knowledge Graph-based Recommendation via Generative Denoising and Multi-Level Contrastive Learning. Data Intelligence, 2026, Vol. 8, Art. No.: 20250099. DOI: https://doi.org/10.3724/2096-7004.di.2025.0099

ABSTRACT

Integrating knowledge graphs into recommendation systems poses significant challenges, particularly in handling noise and distribution mismatch between the knowledge graph and the user-item interaction graph. To address these challenges, we propose a knowledge graph recommendation framework that combines generative denoising with multi-level contrastive learning. This integration of complementary self-supervised objectives alleviates the lack of supervision and enhances the model's ability to learn from sparse user-item interactions. Specifically, to handle the high heterogeneity of knowledge graphs, our method employs a heterogeneous graph Transformer for representation learning. To suppress noise in the graph structure, we introduce a credible knowledge masking and reconstruction mechanism. Furthermore, for fine-grained user preference modeling, we leverage decoupling strategies and multi-preference contrastive learning to capture multi-dimensional interests, enhancing the interpretability of user behavior. Extensive experiments demonstrate that our method surpasses the state-of-the-art baselines.

1. INTRODUCTION

The rapid advancement of Internet technologies has fueled exponential growth in the global digital economy, triggering an unprecedented surge in data generation. This data explosion has intensified the

*Corresponding author: Yi Zhao (E-mail: zhaoyi.hb@gmail.com; ORCID: 0000-0002-0919-867X).

© 2026 Chinese Academy of Sciences. Published under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

challenge of information overload, where users are overwhelmed by vast amounts of content, much of which is irrelevant. Recommendation systems have emerged as a key solution to this problem, leveraging user preferences and item characteristics to deliver personalized content from massive information spaces [1-4].

However, traditional collaborative filtering (CF) methods often suffer from limited generalization capabilities when faced with challenges such as data sparsity and cold start problems, which significantly hinder the performance improvement of recommendation systems. With the rapid advancement and widespread adoption of knowledge graph technologies across various domains, increasing research efforts have focused on leveraging the rich semantic and structural information within knowledge graphs to enhance the effectiveness of recommendation systems [5]. For example, embedding-based [6-8] and path-based [9-10] approaches have been widely explored in early studies. However, these methods either suffer from limited expressive power or rely heavily on manually defined meta-paths. Currently, Graph Neural Networks (GNNs)-based methods [11] are the most prevalent. They learn high-order entity representations by propagating and aggregating information across graph structures, showing strong potential in knowledge-aware recommendation tasks. Despite their effectiveness, existing GNN-based models still face several key challenges. First, they struggle to model the high heterogeneity in knowledge graphs, where nodes and relations are of diverse types. Second, most models assume users' preferences are single-dimensional, failing to capture the multi-faceted nature of real-world user interests. Lastly, knowledge graphs often contain noisy triples that are irrelevant to the recommendation task, which can mislead model training and degrade both accuracy and robustness.

To address the above three issues, this paper proposes a novel recommendation method: a knowledge graph recommendation model GDMCL (Generative Denoising and Multi-Level Contrastive Learning). First, a heterogeneous graph Transformer is utilized to solve the high heterogeneity of knowledge graphs. By introducing a type-aware attention mechanism, it dynamically captures the semantic differences between different node types and relation types, thereby more effectively modeling long-tail entities and complex relations. Second, the problem of insufficient fine-grained modeling of user preferences is addressed through a designed multi-channel decoupling mechanism. The embedding representations of the knowledge graph and user-item interaction graph are respectively mapped to multiple independent semantic spaces to model users' different preference features and entities' diversified attribute features, thus enabling more refined modeling of user preferences and improving the personalization degree of recommendations. Third, a masked autoencoder is introduced to effectively solve the problem of noisy knowledge commonly existing in knowledge graphs. Adaptive masking and reliable knowledge reconstruction are used to filter noisy edges, thereby enhancing the robustness of the recommendation model. Finally, enhanced knowledge views and enhanced collaborative views are constructed, and two contrastive learning mechanisms are introduced to build contrastive learning tasks at the two levels of multi-preference and multi-view. The multi-preference contrastive learning aims to strengthen the feature modeling of users and items in different semantic dimensions from a fine-grained perspective, capturing users' detailed preferences in each dimension. The multi-view contrastive learning aims to align the embedding representations from different views at a macro level, promoting cross-view knowledge sharing.

Our main contributions are as follows:

- We adopt a perceptual attention mechanism to dynamically capture the semantic differences between different node types and relationship types, thereby more effectively modeling long-tail entities and complex relationships.
- We introduce multi-level contrastive learning to jointly promote subspace disentanglement and cross-view semantic alignment, capturing fine-grained semantics.
- We introduce a masked autoencoder to address noisy knowledge in knowledge graphs. Adaptive masking and credible reconstruction filter noisy edges, boosting model robustness.
- Extensive experiments demonstrate that our method consistently outperforms state-of-the-art baselines.

2. RELATED WORK

Traditional recommendation algorithms are primarily categorized into three types: collaborative filtering, content-based recommendation, and hybrid recommendation approaches. Collaborative filtering relies solely on user behavior data, predicting user preferences through similarities between users or items. Sarwar et al. [12] compared various item similarity computation techniques in ItemCF, while Herlocker et al. [13] proposed a set of performance evaluation metrics for collaborative filtering systems based on UserCF. Rendle et al. [1] introduced the Factorization Machine (FM), which extends matrix factorization by modeling high-order feature interactions, significantly improving recommendation performance. However, collaborative filtering still faces challenges such as data sparsity and cold-start problems. Content-based recommendation algorithms analyze item features to generate recommendations, offering advantages in addressing new-item cold-start scenarios. Nevertheless, these methods often produce less diverse recommendations and struggle with new-user cold-start issues due to insufficient interaction data (e.g., Reddy et al. [14] in movie recommendation). Hybrid recommendation algorithms combine multiple techniques to compensate for individual limitations. Notable examples include Google's Wide & Deep [15], which integrates the memorization capability of linear models with the generalization power of deep neural networks, enhancing both accuracy and diversity. DeepFM [16] further improves upon this by replacing the Wide component with a factorization machine.

Knowledge graphs were first introduced by Google to enhance semantic search capabilities. In recent years, their rich semantic information and strong relational reasoning abilities have made them increasingly popular in recommendation systems. They provide new solutions for alleviating data sparsity and cold-start issues, while also improving model interpretability. Current knowledge graph-based recommendation methods can be broadly classified into three categories: embedding-based methods [6, 17], path-based methods [9, 18-19], and graph neural network-based methods [11, 20-23]. Embedding-based methods employ knowledge graph embedding techniques to map entities and relations into low-dimensional dense vectors, which are then integrated into recommendation modules to enrich the semantic representations of users and items. However, these methods typically fail to fully exploit the semantic information contained in connection paths, limiting the interpretability of recommendations. Path-based methods construct meta-paths or meta-graph topologies to uncover latent associations between users and items, aiding the

recommendation process. Despite their effectiveness, these methods suffer from high computational complexity in path selection and require extensive domain knowledge for manual meta-path design, restricting their efficiency and cross-domain adaptability. Graph neural network-based methods leverage the message-passing mechanism of GNNs to capture complex relationships and high-order dependencies among entities in the knowledge graph, generating high-quality embeddings for downstream recommendation tasks.

Although existing knowledge graph-based recommendation algorithms have demonstrated significant improvements in performance, they still face two key challenges: (1) the dominant supervised learning paradigm suffers from sparse supervision signals, and (2) real-world knowledge graphs are often sparse and noisy, increasing training difficulty and degrading recommendation quality. To address these issues, we propose a Knowledge Graph Recommendation Algorithm Based on Generative Denoising and Multi-Level Contrastive Learning (GDMCL), aiming to enhance model robustness and recommendation effectiveness.

3. PROBLEM FORMULATION

GDMCL is a recommendation method based on knowledge graphs, where the used data includes user-item interaction data and knowledge graph data from the item side, and it models these as two graph datasets G_{CF} and G_{KG}. First, for the user-item interaction data, assuming there are M users and N items, the user set and item set are defined as $\mathbf{U} = \{u_1, u_2, ..., u_M\}$ and $\mathbf{I} = \{i_1, i_2, ..., i_N\}$, respectively. The user-item interaction matrix is defined as $Y \in \mathbb{R}^{M \times N}$, where the entry y_{ui} in the u-th row and i-th column represents the interaction record between user u and item i. If $y_{ui} = 1$, it indicates that the user has interacted with the item, such as purchasing, clicking, or viewing, if $y_{ui} = 0$, it indicates no interaction. The user-item interaction data is modeled as a bipartite graph $G_{CF} : \{(u, y_{ui}, i) \mid u \in \mathbf{U}, i \in \mathbf{I}\}$. Second, the knowledge graph on the item side provides additional information about the items. This information typically includes item attributes (such as the director, actors, genre, and type of movies) and other external knowledge. Generally, the knowledge graph is modeled as a set of triplets, formally defined as $G_{KG} : \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ denotes the entity set and $\mathcal{R}$ denotes the relation set. Each triplet describes the connection between head entity h and tail entity t through relation r. Inspired by the modeling approach of KGAT, this paper similarly encodes user behavior and knowledge graphs into a unified collaborative knowledge graph $G_{CKG} : \{(h, r, t) \mid h, t \in \mathcal{E}', r \in \mathcal{R}'\}$, where $\mathcal{E}' = \mathcal{E} \bigcup \mathbf{U}$ and $\mathcal{R}' = \mathcal{R} \bigcup \{\text{Interact}\}$.

4. THE PROPOSED APPROACH

The core work of this chapter is to train GDMCL, utilize generative tasks for self-supervised removal of knowledge noise, learn knowledge-enhanced user and item representations, and predict whether user u has potential interest in item v that has not been interacted with.

4.1 Main Architecture of GDMCL Framework

The overall model architecture of GDMCL is shown in Figure 1, which mainly consists of four modules: the decoupled representation initialization, dual-view graph aggregation module, the generative-denoising module, and the multi-level contrastive learning module.

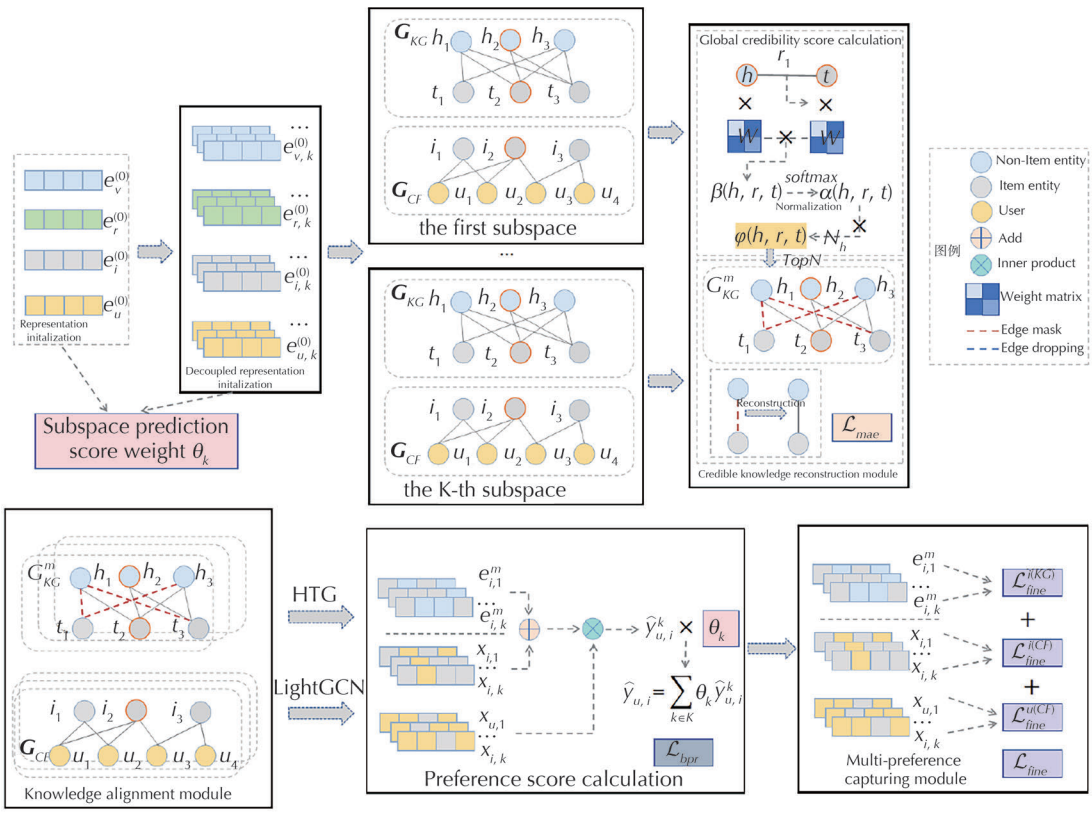

Figure 1. GDMCL Model Architecture Diagram.

GDMCL constructs graphs using user-item interaction and item-side knowledge graph data: interaction data is modeled as a bipartite graph $G_{CF}: \{(u, y_{ui}, i) | u \in U, i \in I\}$ where $y_{ui} = 1/0$ indicates presence/absence of interaction, the item-side knowledge graph is a triple set $G_{KG}: \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$ with entities $\mathcal{E}$ and relations $\mathcal{R}$, inspired by [11], a collaborative knowledge graph $G_{CKG}: \{(h, r, t) | h, t \in \mathcal{E}', r \in \mathcal{R}'\}$ is formed with $\mathcal{E}' = \mathcal{E} \bigcup U$ and $\mathcal{R}' = \mathcal{R} \bigcup \{\text{Interact}\}$.

In addition, during model training, a multi-task joint training approach is adopted, jointly training the recommendation task loss, the multi-preference contrastive learning loss, the multi-view contrastive learning loss, and the masked reconstruction loss.

4.2 Decoupled Representation Initialization

The decoupled representation module aims to achieve fine-grained modeling of user preferences through multi-perspective semantic decoupling, while accurately capturing the multi-faceted semantic information of entities in the knowledge graph. Inspired by DCLKR [24], we adopt the same strategy to initialize the decoupled representations used in subsequent graph aggregation operations. Specifically, a multi-view gating unit is employed to project entities and relations in the knowledge graph view into K subspaces, resulting in the initial decoupled entity embeddings $\mathbf{e}_{v,k}$ and relation embeddings $\mathbf{e}_{r,k}$. Similarly, users and items in the user-item interaction graph (collaborative graph) view are also mapped into K subspaces, yielding the initial decoupled user embeddings $\mathbf{e}_{u,k}$. The above process can be formalized as Eq. 1.

$$\mathbf{e}_{x,k} = \mathbf{e}_x \odot g(\mathbf{W}_x \cdot \mathbf{e}_x + \mathbf{b}_x) \tag{1}$$

where $\mathbf{e}_x \in \{\mathbf{e}_v, \mathbf{e}_r, \mathbf{e}_u\}$ represents the original embedding of entities (v), relations (r), or user IDs (u), and $\mathbf{W}_x$, $\mathbf{b}_x$ are the corresponding learnable parameters. The operator $\odot$ denotes the Hadamard product, and $g(\cdot)$ is the Sigmoid activation function.

4.3 Dual-view Graph Aggregation Module

To better capture the structural characteristics of the knowledge graph view and the collaborative graph view, we apply different types of Graph Convolutional Networks (GCNs) independently on each view. Given the high heterogeneity of knowledge graphs, where various relation types contribute differently to entity representations, traditional GNNs such as GCN [25] and GAT [26] cannot distinguish their importance.

Specifically, as for the knowledge graph view, we introduce a heterogeneous graph transformer in each subspace to dynamically model the impact of different relations. For each triplet (h, r, t) in the knowledge graph, we compute a relation-aware attention score $\beta(h, r, t)$ using an attention mechanism:

$$\beta(h, r, t) = \frac{\mathbf{e}_h \mathbf{W}_Q \cdot (\mathbf{e}_r \odot \mathbf{e}_t \mathbf{W}_K)^\top}{\sqrt{d}} \tag{2}$$

where $\mathbf{W}_Q$ and $\mathbf{W}_K$ are learnable parameter matrices, and $\sqrt{d}$ is a scaling factor. The Hadamard product $\odot$ captures the interaction between relation r and tail entity t, enabling adaptive adjustment of relational importance across semantic spaces. To ensure fair comparison among neighbors of the same head entity, we normalize the scores using the softmax function:

$$\alpha(h, r, t) = \frac{\exp(\beta(h, r, t))}{\sum_{(h, r', t') \in \mathcal{N}_h} \exp(\beta(h, r', t'))} \tag{3}$$

Using these normalized weights, we update the embedding of entity h in the $(l + 1)$-th layer of the k-th subspace as:

$$\mathbf{e}_{h,k}^{(l+1)} = \frac{1}{|\mathcal{N}_h|} \sum_{(h,r,t) \in \mathcal{N}_h} \alpha(h,r,t)\, \mathbf{e}_{r,k} \odot \mathbf{e}_{t,k}^{(l)} \tag{4}$$

For the collaborative graph view, we generate user embeddings by aggregating neighboring item embeddings. We adopt LightGCN [27] to refine the collaborative view. The updated representation for node $x \in \{u, i\}$ is given by:

$$\boldsymbol{x}_{x,k}^{(l+1)} = \sum_{y \in \mathcal{N}_x} \frac{\boldsymbol{x}_{y,k}^{(l)}}{\sqrt{|\mathcal{N}_x||\mathcal{N}_y|}} \tag{5}$$

where $\mathcal{N}_x$ denotes the set of neighbors of node x in the user-item interaction graph. Finally, we aggregate multi-layer embeddings for entities v, users u, and items i in each subspace using:

$$\boldsymbol{z}_{t,k} = \mathrm{AGG}\left\{\boldsymbol{z}_{t,k}^{(0)}, \boldsymbol{z}_{t,k}^{(1)}, \ldots, \boldsymbol{z}_{t,k}^{(L)}\right\} \tag{6}$$

where AGG(·) denotes the aggregation function (e.g., mean or sum), and L is the total number of layers.

4.4 Generative-denoising Module

In this section, we present a generative denoising module, where a batch of knowledge triples is randomly masked, and the model is trained to reconstruct the masked relations from the corrupted inputs. This serves as a generative self-supervised objective to enhance robust representation learning.

First, we compute the attention scores in Eq. 3, which reflect the local importance of triplets among all edges of the head entity h. However, the degree of the head entity may affect the global importance evaluation of triplets within the entire knowledge graph. To address this issue, we multiply the original attention scores by the number of head entity neighbors $|\mathcal{N}_h|$, resulting in the global credibility score $\varphi(h, r, t)$, as shown in Eq. 7.

$$\varphi(h,r,t) = |\mathcal{N}_h| \cdot \alpha(h,r,t) = \frac{|\mathcal{N}_h| \cdot \exp(\beta(h,r,t))}{\sum_{(h,r',t') \in \mathcal{N}_h} \exp(\beta(h,r',t'))} \tag{7}$$

To further enhance the robustness of sampling, we add Gumbel noise to the learned global credibility score $\varphi(h, r, t)$, as shown in Eq. 8.

$$\varphi(h,r,t) = \varphi(h,r,t) - \log(-\log(\epsilon)), \quad \epsilon \sim \mathrm{Uniform}(0,1) \tag{8}$$

Here, ϵ is a random variable sampled from a uniform distribution. After adding Gumbel noise to the global credibility score, we select the top n triplets with the highest scores for masking. The masking process is shown in Eq. 9.

$$\mathcal{M}_n = \{(h,r,t) \mid \varphi(h,r,t) \in \mathrm{topn}(\Gamma, n_m)\}, \quad \mathbf{G}_{KG} - \mathcal{M}_n \tag{9}$$

Here, Γ represents the distribution of all $\varphi(h, r, t)$. Removing M_n from the original knowledge view graph G_{KG} results in an enhanced knowledge view graph G_{KG}^m with low-confidence triplets after deleting high-confidence ones.

After obtaining the masked enhanced knowledge view graph G_{KG}^m, we use Eq.4 to update the representation $\boldsymbol{e}_{v,k}^{m(l)}$ of entity e in the l-th layer of the k-th subspace. After multiple layers of graph convolution, we aggregate information using sum and aggregator operations to obtain the final embedding $\boldsymbol{e}_{v,k}^{m}$ of entity e in the enhanced view graph G_K^m, as shown in Eq. 10.

$$\mathbf{e}_{v,k}^{m} = \mathrm{AGG}\left\{\mathbf{e}_{v,k}^{m(0)}, \mathbf{e}_{v,k}^{m(1)}, \ldots, \mathbf{e}_{v,k}^{m(l)}\right\} \tag{10}$$

The masked knowledge triplet set $\mathcal{M}_n$ has a high global credibility score. In the enhanced view, due to being masked, these triplets are not used during graph aggregation and can serve as self-supervised labels for reconstructing credible knowledge. We obtain the head entity embedding $\boldsymbol{e}_{h,k}^{m}$ and the tail entity embedding $\boldsymbol{e}_{t,k}^{m}$ based on the entity index, and the reconstruction loss is shown in Eq. 11.

$$\mathcal{L}_{mae} = \sum_{(h,r,t)\in\mathcal{M}_n} -\log\left(g(\mathbf{e}_{h,k}^{m\top}\cdot(\mathbf{e}_{r,k}\odot\mathbf{e}_{t,k}^{m}))\right) \tag{11}$$

Here, $g(\cdot)$ denotes the Sigmoid activation function, $\odot$ represents the dot product operation between the tail entity and the relation.

4.5 Multi-level Contrastive Learning Module

To promote independence among disentangled representations and capture diverse node features, we employ intra-view contrastive learning. In this setting, representations of the same node within the same view and subspace are treated as positive samples, while those from different subspaces serve as negatives.

In the knowledge graph view, we optimize entity embeddings independently for each subspace. As shown in Eq. 12, we use the InfoNCE loss to maximize the similarity of positive pairs and minimize that of negative pairs across subspaces.

$$\mathcal{L}_{fine}^{v(KG)} = \sum_{e\in\mathcal{E}_k}\sum_{k\in\mathcal{K}} -\log\frac{\exp(s(\mathbf{e}_{v,k}, \mathbf{e}_{v,k})/\tau_1)}{\sum_{k'\in\mathcal{K}}\exp(s(\mathbf{e}_{v,k}, \mathbf{e}_{v,k'})/\tau_1)} \tag{12}$$

Here, $\mathbf{e}_{v,k}$ denotes the aggregated embedding representation of entity v in the masked knowledge view graph G_{KG}^m. The function $s(\cdot)$ represents the cosine similarity, and τ_1 is a temperature parameter used to adjust the scaling of the similarity.

As for the collaborative view, the InfoNCE loss function is also employed to minimize the distance between positive samples while maximizing the distance between negative samples. The fine-grained contrastive learning loss is applied to both users and items in the collaborative view graph G_{CF}, aiming to distinguish different subspaces of the same user or item from others. We unify the formulation for users and items as follows:

$$\mathcal{L}_{fine}^{i(CF)} = \sum_{v \in V} \sum_{k \in K} -\log \frac{\exp(s(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k}) / \tau_1)}{\sum_{k' \in K} \exp(s(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{i,k'}) / \tau_1)} \tag{13}$$

$$\mathcal{L}_{fine}^{u(CF)} = \sum_{u \in U} \sum_{k \in K} -\log \frac{\exp(s(\boldsymbol{x}_{u,k}, \boldsymbol{x}_{u,k}) / \tau_1)}{\sum_{k' \in K} \exp(s(\boldsymbol{x}_{u,k}, \boldsymbol{x}_{u,k'}) / \tau_1)} \tag{14}$$

where $\boldsymbol{x}_{u,k}$ and $\boldsymbol{x}_{i,k}$ represent the embedding of a user or item in the k-th subspace within G_{CF}. The overall intra-view contrastive learning loss is defined as follows.

$$\mathcal{L}_{fine} = \mathcal{L}_{fine}^{e(KG)} + \mathcal{L}_{fine}^{i(CF)} + \mathcal{L}_{fine}^{u(CF)} \tag{15}$$

Subsequently, the InfoNCE loss function is introduced to perform representation alignment across view. In this setting, the same item from different views is treated as a positive sample, while different items from different views are considered negative samples. The contrastive learning loss is defined as follows.

$$\mathcal{L}_{align} = \frac{1}{|K|} \sum_{k \in \mathcal{K}} \sum_{v \in \mathcal{V}} -\log \frac{\exp\left(\frac{s(\mathbf{e}_{v,k}^{(CF)}, \mathbf{e}_{v,k}^{(KG)})}{\tau_2}\right)}{\exp\left(\frac{s(\mathbf{e}_{v,k}^{(CF)}, \mathbf{e}_{v,k}^{(KG)})}{\tau_2}\right) + \sum_{v'} \exp\left(\frac{s(\mathbf{e}_{v',k}^{(CF)}, \mathbf{e}_{v',k}^{(KG)})}{\tau_2}\right)} \tag{16}$$

Here, K denotes the number of subspaces, τ_2 represents the temperature parameter, which is used to adjust the scaling of similarity, and $s(\cdot)$ denotes the cosine similarity.

4.6 Multi-task Joint Optimization

The proposed model jointly optimizes the recommendation task, the generative denoising task, and the multi-level contrastive learning task. For the main recommendation objective, predictions are made using the dot product between user and item embeddings. To train the model, the Bayesian Personalized Ranking loss is adopted to update parameters, as defined in the following:

$$\mathcal{L}_{bpr} = \sum_{(u,i,j) \in \mathcal{O}} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \tag{17}$$

Here, $\hat{y}_{ui}$ and $\hat{y}_{uj}$ are the predicted scores for user u and items i and j, respectively. $\mathcal{O} = \{(u, i, j) \mid (u, i) \in \mathcal{O}^+, (u, j) \in \mathcal{O}^-\}$, where $\mathcal{O}^+$ represents observed interactions, and $\mathcal{O}^-$ represents unobserved interactions.

Since GDMCL is decoupled and separate, different subspaces can provide distinct perspectives to capture the relationships between users and items. In this section, we introduce an attention fusion strategy that can effectively distinguish the advantages of each subspace and deeply explore users' true preferences. It is necessary to calculate the prediction score $\hat{y}_{ui}^k$ for each subspace and weight it according to its fusion weight. The final overall prediction score $\hat{y}_{ui}$ is obtained by summing these weighted scores. The prediction score for each subspace is calculated as shown in Eq.18.

$$\begin{aligned} \boldsymbol{z}_{i,k} &= \mathbf{e}^{m}_{i,k} \,\|\, \boldsymbol{x}_{i,k}, \quad \boldsymbol{z}_{u,k} = \mathbf{e}^{m}_{u,k} \,\|\, \boldsymbol{x}_{u,k} \\ \hat{y}^{k}_{ui} &= \boldsymbol{z}^{\top}_{u,k} * \boldsymbol{z}_{i,k} \end{aligned} \tag{18}$$

$$\theta_k = \frac{\exp(\mathrm{MLP}(\mathbf{e}_u \,\|\, \mathbf{e}_i)^{\top} \mathrm{MLP}(\mathbf{e}_{u,k} \,\|\, \mathbf{e}_{i,k}))}{\sum_{k' \in \mathcal{K}} \exp(\mathrm{MLP}(\mathbf{e}_u \,\|\, \mathbf{e}_i)^{\top} \mathrm{MLP}(\mathbf{e}_{u,k'} \,\|\, \mathbf{e}_{i,k'}))} \tag{19}$$

$$\hat{y}_{ui} = \sum_{k \in \mathcal{K}} \theta_k \hat{y}^{k}_{ui} \tag{20}$$

Here, $\boldsymbol{e}^{m}_{i,k}$ and $\boldsymbol{e}^{m}_{u,k}$ are the item embedding and user embedding in the masked augmented view, respectively. $\boldsymbol{x}_{i,k}$ and $\boldsymbol{x}_{u,k}$ are the item embedding and user embedding in the collaborative view, respectively. $\mathbf{e}_u$ and $\mathbf{e}_i$ are the initialized user embedding and item embedding, respectively. $\mathbf{e}_{u,k}$ and $\mathbf{e}_{i,k}$ are the initialized user decoupled embedding and item decoupled embedding, respectively.

Finally, the multi-task joint contrastive learning loss is employed to optimize GDMCL, and the corresponding formula is defined as follows.

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda_1 \mathcal{L}_{fine} + \lambda_2 \mathcal{L}_{align} + \lambda_3 \mathcal{L}_{mae} \tag{21}$$

5. EXPERIMENTS AND ANALYSIS

5.1 Datasets

To evaluate the effectiveness of the GDMCL algorithm, this study selects three benchmark datasets from different domains for experiments: Last-FM (music recommendation), Book-Crossing (book recommendation), and Movielens-1M (movie recommendation). These datasets exhibit significant differences in scale and interaction sparsity, enabling an effective examination of the adaptability and generalization ability of GDMCL in various application scenarios. Table 1 lists the statistical characteristics of the datasets used in the experiments.

- Last-FM: A music dataset constructed by the online music platform last.fm based on user behavior data, which includes the attribute information of 1,872 users and 3,846 songs. Each song is regarded as an item, and a user's listening to a song is considered an interaction. To distinguish it from the Last-FM used in the LCLSA model, it is referred to as Last.fm-2K here.
- Book-Crossing: Records user rating information in the book domain, containing rating data for 14,967 books, involving 17,860 users and 14,967 items. The rating scale ranges from 0 to 10.
- Movielens-1M: Originating from the well-known movie platform Movielens, it is a widely-used benchmark dataset in the field of movie recommendation. This dataset includes the rating records of 6,036 users for 2,445 movies. It adopts a 5-level rating system, that is, the rating scale ranges from 1 to 5.

Table 1. Data Statistics of Datasets for GDMCL Experiments.

Parameter	Last-FM	Book-Crossing	Movielens-1M
Number of Users	1872	17860	6036
Number of Items	3846	14967	2445
Number of Interactions	42346	139746	753772
Interaction Sparsity	5.9×10^{-3}	5.2×10^{-4}	5.1×10^{-2}
Number of Entities	6366	77903	182011
Number of Relations	60	25	12
Number of Triples	15518	151500	1241996

Since the above three datasets are explicit rating data, following the processing method of RippleNet, the explicit ratings are converted into implicit ratings. For the Last-FM and Book-Crossing datasets, considering their sparsity, as long as a user rates or interacts with an item, it is regarded as a positive sample, and negative samples are randomly selected from the items that the user has not interacted with, with the same number as the positive samples. For the Movielens-1M dataset, samples with a rating greater than 4 are selected as positive samples. The auxiliary information of the knowledge graph is obtained from Satori, a Microsoft commercial knowledge graph. In the experiment, the datasets are divided into an 80% training set and a 20% test set.

5.2 Baseline models and Settings

Baseline models: To evaluate the performance of the GDMCL model on the Click-Through Rate (CTR) task and the Top-K recommendation task, we compare it with five categories of baseline models: methods without knowledge graphs (BPRMF [17]), and four categories of methods all based on knowledge graphs—embedding-based methods (CKE [28], MKR [29]), path-based methods (RKGE [18], RippleNet [19]), graph neural network-based methods (KGCN [20], KGAT [11], CKAN [21], KGIN [22], KFGAN [30]), and contrastive learning-based methods (KGIC [31], KGCL [32], KGTN [23], MCCLK [33]).

Settings: To ensure relative fairness in the comparison with baseline methods, the experiment fixes the embedding dimension of all baseline models at 64 and initializes the embedding parameters using the Xavier initialization strategy. With the same embedding dimension, a hyperparameter grid search is conducted for the GDMCL model. The learning rate ranges from {0.1, 0.01, 0.001, 0.001}, the number of graph convolution layers is set to 1, 2, 3, 4, the number of subspaces is {2, 3, 4, 5}, and the mask dimension is {16, 32, 64, 128, 256, 512, 1024}. The ranges of preference contrastive learning weight, knowledge contrastive learning weight, and mask reconstruction weight are all set to {0.001, 0.01, 0.1, 1}.

5.3 Experimental Results

As shown in Table 2, we evaluate the performance of GDMCL and 14 baseline models on the CTR task across three public datasets (Last-FM, Book-Crossing, and Movielens-1M), and we employ two metrics: AUC and F1 score.

Table 2. CTR performance comparison of GDMCL with baseline models on three public datasets.

Baseline Model	Last-FM		Book-Crossing		Movielens-1M	
	AUC	F1	AUC	F1	AUC	F1
BPRMF	0.7626	0.7083	0.6620	0.6157	0.8903	0.7930
CKE	0.7869	0.6954	0.6861	0.6245	0.9094	0.8394
MKR	0.8166	0.7077	0.7444	0.6154	0.9188	0.8262
RKGE	0.7670	0.7250	0.6890	0.6150	0.8860	0.8150
RippleNet	0.8278	0.7484	0.7424	0.6820	0.9018	0.8282
KGCN	0.8068	0.7331	0.6890	0.6698	0.8895	0.8245
KGAT	0.8293	0.7424	0.7314	0.6544	0.9140	0.8440
CKAN	0.8464	0.7704	0.7417	0.6673	0.9076	0.8403
KGIN	0.8486	0.7602	0.7273	0.6614	0.9190	0.8441
KGIC	0.8569	0.7779	0.7505	0.6706	0.8882	0.8194
KGCL	0.8455	0.7596	0.7453	0.6679	0.9184	0.8437
KFGAN	0.8603	0.7731	0.7795	0.6900	0.9253	0.8567
KGTN	0.8607	0.7745	0.7691	0.6893	0.9313	0.8581
MCCLK	0.8722	0.7864	0.7566	0.6817	0.9341	0.8617
GDMCL	**0.8922**	**0.8073**	**0.7851**	**0.7006**	**0.9431**	**0.8712**
Gain(%)	2.29%	2.66%	0.72%	1.64%	0.96%	1.10%

Overall, GDMCL outperforms all baseline models. Specifically, it achieves improvements of 2.29% in AUC and 2.66% in F1 on Last-FM, 0.72% in AUC and 1.64% in F1 on Book-Crossing, and 0.96% in AUC and 1.10% in F1 on Movielens-1M, demonstrating its overall effectiveness. GDMCL also shows superior recall performance. The key reasons for its improved performance are as follows: First, unlike models such as KGIN, KGAT, KGTN, and MCCLKG, GDMCL introduces a knowledge denoising mechanism that addresses the issue of noisy triples in knowledge graphs. Existing models lack such a mechanism, making it difficult to distinguish useful information from noise, which hinders user and item representation learning.

In contrast, GDMCL enhances learning reliability through trustworthy knowledge reconstruction. GDMCL better captures users' multi-faceted preferences by splitting them into multiple subspaces, enabling fine-grained modeling from different perspectives and leading to a more accurate depiction of user interests.

Second, baseline models that integrate knowledge graphs, including GDMCL, generally outperform BPRMF, which does not use knowledge graphs, in the CTR task. This validates the effectiveness of incorporating knowledge graphs into recommendation systems and highlights their practical value. Moreover, GNN-based methods typically outperform embedding-based and path-based approaches, indicating that GNNs better exploit structural and high-order information in knowledge graphs.

However, KGCN underperforms RippleNet in some cases, suggesting that excessive noise or redundant neighbor information in knowledge graphs may degrade the performance of GNN-based aggregation. In contrast, RippleNet employs a path-based interest diffusion mechanism with attention to filter high-correlation paths, enabling more accurate user preference modeling. This further confirms the importance of effective information filtering in knowledge graph-enhanced recommendation systems.

5.4 Ablation Experiments

To evaluate the contribution of each component in GDMCL, we conduct ablation studies by removing four core modules and comparing AUC and F1 scores across three datasets. The variants are described as follows: **-AttnHGT**: Replaces the heterogeneous graph Transformer with a standard relational path-aware convolutional network to assess the impact of the knowledge graph aggregation module. **-MAE**: Removes the masked auto-encoder to disable trustworthy knowledge reconstruction, evaluating its role in filtering noisy knowledge. **-IntraViewCL**: Removes the decoupled representation module, simplifying user preferences to a single view, to verify the benefit of modeling multi-faceted preferences. **-CrossViewCL**: Removes the multi-subspace contrastive learning mechanism, assessing the impact of fine-grained preference capture.

The ablation results in Table 3 show that removing the heterogeneous graph Transformer (-AttnHGT) leads to a significant performance drop across all datasets. This is because traditional relational path-aware convolutional networks fail to capture global semantics and integrate heterogeneous information effectively, limiting the model's ability to learn deep associations between users and items. Removing the trustworthy knowledge reconstruction module (-MAE) disables the model's ability to filter out noisy triples, resulting in interference from irrelevant knowledge and a consequent decline in performance. Removing the decoupled representation module (-MultiView) forces the model to represent user preferences as a single view, failing to capture their multi-faceted nature. This reduces the diversity and accuracy of recommendations. Removing the multi-preference capture module (-FineCL) weakens the model's ability to distinguish and model fine-grained user preferences, further degrading recommendation quality. These results demonstrate that each core component of GDMCL contributes uniquely to the overall performance, and their integration ensures effective and comprehensive modeling of user interests and item features.

Table 3. Results of ablation experiments of GDMCL on three public datasets.

Model	Last-FM		Book-Crossing		Movielens-1M	
	AUC	F1	AUC	F1	AUC	F1
GDMCL	0.8922	0.8073	0.7851	0.7006	0.9431	0.8712
-AttnHGT	0.8868	0.8048	0.7784	0.6914	0.9384	0.8682
-MAE	0.8880	0.8035	0.7802	0.6927	0.9404	0.8676
-IntraViewCL	0.8914	0.8072	0.7822	0.6936	0.9387	0.8698
-CrossViewCL	0.8869	0.8017	0.7799	0.6931	0.9386	0.8695

5.5 Hyperparameter Analysis

To better understand the role of key components in GDMCL, we analyze the impact of two contrastive learning weights: the multi-preference learning weight (λ_1), knowledge alignment weight (λ_2). First, we evaluate the effect of λ_1 and λ_2 on model performance by varying their values within the range {0.001, 0.01, 0.1, 1}, while keeping other parameters fixed. The results for different λ_1 settings are summarized in Table 4, and those for λ_2 in Table 5. From Table 4, we observe that increasing λ_1 initially improves performance, with the best results achieved at $\lambda_1 = 0.01$ on Last-FM and $\lambda_1 = 0.1$ on Book-Crossing and Movielens-1M. Further increases lead to a decline in performance. Similarly, Table 5 shows that $\lambda_2 = 0.1$ yields optimal results

Table 4. Performance changes of GDMCL under different preference contrastive learning weights (λ_1).

λ_1	Last-FM		Book-Crossing		Movielens-1M	
	AUC	F1	AUC	F1	AUC	F1
0.001	0.8901	0.8038	0.7824	0.6936	0.9415	0.8672
0.01	0.8922	0.8073	0.7820	0.6954	0.9423	0.8694
0.1	0.8862	0.7980	0.7851	0.7006	0.9431	0.8712
1	0.8785	0.7934	0.7752	0.6933	0.9407	0.8678

Table 5. Performance changes of GDMCL under different knowledge-alignment contrastive learning weights (λ_2).

λ_2	Last-FM		Book-Crossing		Movielens-1M	
	AUC	F1	AUC	F1	AUC	F1
0.001	0.8908	0.8067	0.7849	0.6922	0.9401	0.8696
0.01	0.8922	0.8073	0.7851	0.7006	0.9431	0.8712
0.1	0.8899	0.8066	0.7848	0.6944	0.9409	0.8688
1	0.8831	0.7974	0.7826	0.6949	0.9379	0.8636

across all datasets, beyond which performance deteriorates. These findings suggest that moderate contrastive learning weights enhance model training by balancing auxiliary and main tasks. However, excessively large weights may disrupt the optimization of the primary recommendation objective.

6. CONCLUSION AND FUTURE WORK

This paper presents GDMCL, a novel knowledge graph recommendation model that integrates generative denoising and multi-level contrastive learning. By combining generative and contrastive self-supervised paradigms, GDMCL addresses sparse supervision and improves the extraction of informative knowledge signals. To tackle knowledge graph heterogeneity, we introduce a heterogeneous graph Transformer to dynamically capture complex relational structures, replacing manual relation-aware encodings. For noise suppression, we propose a generative self-supervised task based on trustworthy knowledge masking and reconstruction, effectively filtering irrelevant information. Additionally, to model fine-grained user preferences, we adopt a decoupled representation framework with multi-perspective contrastive learning, capturing nuanced interests and enhancing personalization and interpretability. Experiments on multiple real-world datasets confirm GDMCL's superior performance across domains. This study has limitations: first, insufficient utilization of user-side information, focusing on enhancing item embedding via item-side knowledge graphs while failing to fully explore the potential of user-side information in improving recommendation performance, second, unresolved cold-start issue—though knowledge graph auxiliary information alleviates it to some extent, the model's recommendation performance remains limited for new items or users with no interaction records.

AUTHOR CONTRIBUTIONS

Jibing Gong: Supervision; Writing-Review & Editing; Funding Acquisition; Project Administration; Resources;

Xinchao Feng: Methodology; Investigation; Writing-Original Draft; Data Curation; Validation; Formal Analysis; Software;

Qian Zang: Conceptualization; Investigation; Writing-Original Draft; Data Curation; Validation; Resources;

Yi Zhao: Supervision; Project Administration; Writing-Review & Editing;

Le Chen: Visualization; Data Curation; Formal Analysis; Validation;

Yu Qiu: Investigation;

Jun Feng: Formal Analysis; Visualization;

Haomin Song: Writing-Original Draft; Investigation;

Liangying Peng: Writing-Original Draft; Visualization.

ACKNOWLEDGEMENTS

This work was supported by resources provided by Central Government Guidance Local Science and Technology Development Fund Project (246Z0306G) and Innovation Capability Improvement Plan Project of Hebei Province (22567626H).

REFERENCES

[1] S. Rendle, "Factorization machines," *2010 IEEE International Conference on Data Mining (ICDM)*, pp. 995–1000, 2010. https://doi.org/10.1109/ICDM.2010.127.

[2] J. Gong *et al*., "Personalized recommendation via inductive spatiotemporal graph neural network," *Pattern Recognition*, vol. 145, Art. no. 109884, 2024. https://doi.org/10.1016/j.patcog.2023.109884.

[3] Y. Zhao *et al*., "Beyond the overlapping users: Cross-domain recommendation via adaptive anchor link learning," *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*, pp. 1488–1497, 2023. https://doi.org/10.1145/3539618.3591642.

[4] Y. Zhao *et al*., "Cross-domain recommendation via adaptive bi-directional transfer graph neural networks," *Knowledge and Information Systems*, vol. 67, no. 1, pp. 579–602, 2025. https://doi.org/10.1007/s10115-024-02246-9.

[5] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," *Proceedings of the 2018 World Wide Web Conference (WWW '18)*, pp. 1835–1844, 2018. https://doi.org/10.1145/3178876.3186175.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems 26 (NeurIPS 2013)*, pp. 2787–2795, 2013. https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[7] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14)*, pp. 1112–1119, 2014. https://doi.org/10.1609/aaai.v28i1.8870.

[8] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, pp. 2181–2187, 2015. https://doi.org/10.1609/aaai.v29i1.9491.

[9] X. Yu *et al*., "Personalized entity recommendation: A heterogeneous information network approach," *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14)*, pp. 283–292, 2014. https://doi.org/10.1145/2556195.2556259.

[10] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 5329–5336, 2019. https://doi.org/10.1609/aaai.v33i01.33015329.

[11] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, pp. 950–958, 2019. https://doi.org/10.1145/3292500.3330989.

[12] B. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*, pp. 285–295, 2001. https://doi.org/10.1145/371920.372071.

[13] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004. https://doi.org/10.1145/963770.963772.

[14] S. R. S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," *Smart Intelligent Computing and Applications*, vol. 105, pp. 391–397, 2019. https://doi.org/10.1007/978-981-13-1927-3_42.

[15] H.-T. Cheng *et al.*, "Wide & deep learning for recommender systems," *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS '16)*, pp. 7–10, 2016. https://doi.org/10.1145/2988450.2988454.

[16] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 1725–1731, 2017. https://doi.org/10.24963/ijcai.2017/239.

[17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, pp. 452–461, 2009. https://www.auai.org/uai2009/papers/UAI2009_0139_48141db02b9f0b02bc7158819ebfa2c7.pdf.

[18] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*, pp. 297–305, 2018. https://doi.org/10.1145/3240323.3240361.

[19] H. Wang *et al.*, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*, pp. 417–426, 2018. https://doi.org/10.1145/3269206.3271739.

[20] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," *Proceedings of the World Wide Web Conference (WWW '19)*, pp. 3307–3313, 2019. https://doi.org/10.1145/3308558.3313417.

[21] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu, "CKAN: Collaborative knowledge-aware attentive network for recommender systems," *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, pp. 219–228, 2020. https://doi.org/10.1145/3397271.3401141.

[22] X. Wang *et al.*, "Learning intents behind interactions with knowledge graph for recommendation," *Proceedings of the Web Conference 2021 (WWW '21)*, pp. 878–887, 2021. https://doi.org/10.1145/3442381.3450133.

[23] D. Zou, W. Wei, F. Zhu, C. Xu, T. Zhang, and C. Huo, "Knowledge enhanced multi-intent transformer network for recommendation," *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion)*, pp. 1–9, 2024. https://doi.org/10.1145/3589335.3648296.

[24] S. Huang, C. Hu, W. Kong, and Y. Liu, "Disentangled contrastive learning for knowledge-aware recommender system," *The Semantic Web – ISWC 2023, Lecture Notes in Computer Science*, vol. 14265, pp. 140–158, 2023. https://doi.org/10.1007/978-3-031-47240-4_8.

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *International Conference on Learning Representations (ICLR 2017)*, 2017. https://openreview.net/forum?id=SJU4ayYgl.

[26] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *International Conference on Learning Representations (ICLR 2018)*, 2018. https://openreview.net/forum?id=rJXMpikCZ.

[27] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, pp. 639–648, 2020. https://doi.org/10.1145/3397271.3401063.

[28] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pp. 353–362, 2016. https://doi.org/10.1145/2939672.2939673.

[29] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," *Proceedings of the World Wide Web Conference (WWW '19)*, pp. 2000–2010, 2019. https://doi.org/10.1145/3308558.3313411.

[30] W. Wang, X. Shen, B. Yi, H. Zhang, J. Liu, and C. Dai, "Knowledge-aware fine-grained attention networks with refined knowledge graph embedding for personalized recommendation," *Expert Systems with Applications*, vol. 249, Art. no. 123710, 2024. https://doi.org/10.1016/j.eswa.2024.123710.

[31] D. Zou *et al*., "Improving knowledge-aware recommendation with multi-level interactive contrastive learning," *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22)*, pp. 2817–2826, 2022. https://doi.org/10.1145/3511808.3557358.

[32] Y. Yang, C. Huang, L. Xia, and C. Li, "Knowledge graph contrastive learning for recommendation," *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*, pp. 1434–1443, 2022. https://doi.org/10.1145/3477495.3532009.

[33] D. Zou *et al*., "Multi-level cross-view contrastive learning for knowledge-aware recommender system," *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*, pp. 1358–1368, 2022. https://doi.org/10.1145/3477495.3532025.