

Article

Adaptive Knowledge Contrastive Learning with Dynamic Attention for Recommender Systems

Hongchan Li, Jinming Zheng, Baohua Jin and Haodong Zhu *

School of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou 450001, China; 2011017@zzuli.edu.cn (H.L.); 332207050653@zzuli.edu.cn (J.Z.); 1988002@zzuli.edu.cn (B.J.)

* Correspondence: zhdzzuliketizu@163.com

Abstract: Knowledge graphs equipped with graph network networks (GNNs) have led to a successful step forward in alleviating cold start problems in recommender systems. However, the performance highly depends on precious high-quality knowledge graphs and supervised labels. This paper argues that existing knowledge-graph-based recommendation methods still suffer from insufficiently exploiting sparse information and the mismatch between personalized interests and general knowledge. This paper proposes a model named Adaptive Knowledge Contrastive Learning with Dynamic Attention (AKCL-DA) to address the above challenges. Specifically, instead of building contrastive views by randomly discarding information, in this study, an adaptive data augmentation method was designed to leverage sparse information effectively. Furthermore, a personalized dynamic attention network was proposed to capture knowledge-aware personalized behaviors by dynamically adjusting user attention, therefore alleviating the mismatch between personalized behavior and general knowledge. Extensive experiments on Yelp2018, LastFM, and MovieLens datasets show that AKCL-DA achieves a strong performance, improving the NDCG by 4.82%, 13.66%, and 4.41% compared to state-of-the-art models, respectively.

Keywords: knowledge graph; recommender systems; dynamic attention; contrastive learning



Citation: Li, H.; Zheng, J.; Jin, B.; Zhu, H. Adaptive Knowledge Contrastive Learning with Dynamic Attention for Recommender Systems. *Electronics* **2024**, *13*, 3594. <https://doi.org/10.3390/electronics13183594>

Academic Editor: George Angelos Papadopoulos

Received: 22 August 2024

Revised: 8 September 2024

Accepted: 8 September 2024

Published: 10 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recommendation systems (RS) alleviate the problem of information overload by providing personalized recommendations and have been extensively deployed in real-life services, such as music platforms [1], video streaming websites [2], and e-commerce [3]. Traditional collaborative filtering (CF) methods [4,5] focus on mining similarities in historical interactions between users and items. These studies achieved significant success across various recommendation services and significantly advanced the field.

Despite advancements from matrix factorization (MF) [6] to graph neural networks (GNNs) [7] in capturing complex user behaviors, CF methods still face the persistent problems of cold start and data sparsity because they rely solely on historical interactions. To address these issues, numerous studies have incorporated knowledge graphs (KGs) [8] into recommendation systems. KGs serve as an external auxiliary resource, providing additional context and demonstrating strong capabilities in relational data modeling. This integration helps enhance the accuracy and explanation of recommendations.

Knowledge-aware methods aim to fully leverage the graphical relationship in KGs and collaborative signals from interaction graphs [9–12]. Early research [13–15] focused on knowledge graph embedding (KGE) techniques, such as TransE [16] and TransH [17], which learn item representations in an embedding space. However, these models often fail to capture global graphical relationships and collaborative signals, as they learn each item independently. Path-based approaches [1,18,19] subsequently utilized KG structural information to elucidate user–item relationships. However, path-based methods are hampered by path-selection issues, typically requiring domain-specific expertise and causing

optimization challenges. Inspired by the strengths of graph neural networks (GNNs), researchers have unified user–item interactions and knowledge graphs into heterogeneous graphs [10,20,21] and adopted information propagation paradigms to capture user behavior patterns. Although these methods achieve promising results, their performance heavily relies on the quality of the KGs, which are often sparse and noisy [22].

Self-supervised contrastive learning (SSL) has shown promise in addressing data sparsity by leveraging relationships within data [23]. However, current approaches to generating contrastive views often rely on techniques such as randomly discarding information, edge perturbation, and subgraph sampling. These methods often result in the problem of insufficiently exploiting sparse information. In addition, existing knowledge graph-based recommendation systems typically employ static attention mechanisms to measure all nodes uniformly. This approach fails to account for the personalized characteristics of different nodes, leading to a mismatch between individual interests and the general knowledge in KGs.

The paper proposes a novel model named Adaptive Knowledge Contrastive Learning with Dynamic Attention (AKCL-DA) to enhance recommender systems by addressing the limitations of sparse information and personalized user behaviors. AKCL-DA employs an adaptive data augmentation strategy on both user–item and item–entity graphs, where hypothetical edges are added to expand the semantic associations within the knowledge graph and comprehensively capture user interests. This method ensures that the contrastive views are more representative of underlying data relationships. Furthermore, a dynamic attention network, inspired by the GATv2 model [24], is designed to capture weight scores between entities and relations, allowing for personalized attention allocation without added computational complexity. This network dynamically adjusts user attention based on personalized behaviors, bridging the gap between individual interests and general knowledge. The essential contributions of this article are as follows:

1. An adaptive data augmentation strategy is proposed to expand semantic associations and user behaviors in knowledge and interaction graphs by adding hypothetical edges. This strategy effectively leverages sparse information, thereby ensuring contrastive views are more representative of underlying data relationships.
2. A personalized dynamic attention network is proposed to bridge the gap between individual interests and general knowledge by dynamically adjusting user attention based on behaviors, alleviating the mismatch found in existing methods.
3. Extensive experiments are conducted on the Yelp2018, LastFM, and MovieLens datasets, and the proposed AKCL-DA method shows significant improvements over state-of-the-art recommendation methods.

2. Related Research

2.1. Knowledge-Aware Recommendation

The recommendation system uses the knowledge graph as supplemental data to achieve good performance in resolving cold start and data sparse issues. The three primary categories of knowledge-aware recommendation techniques now in use are GNN-based, path-based, and embedding-based techniques.

Embedding-based recommendation methods use knowledge-graph-embedding (KGE) technology (for example translation distance model TransE [16] and semantic matching model DistMult [25]) to preprocess KG, and then embedding entities into user and item representations to enhance semantic representation in recommendations. To enter the project's structural, textual, and visual knowledge into the knowledge base embedding module of the CF framework, CKE [12] uses TransR [26] coding. To model news, DKN [11] combines the knowledge-level embedding of entities in news content using TransD, with the text embedding of phrases learned by KCNN [27]. It integrates the representation of news at the semantic and knowledge levels. KTUP [13] proposed joint learning recommendation tasks and knowledge graph completion, using triples to jointly learn user preferences. While embedding-based recommendation algorithms yield better recommendations, their

main advantage is that they represent semantic relevance more accurately, which is more helpful for edge prediction and knowledge graph completion than for recommendation.

Path-based approaches seek to mine possible information between items in the knowledge graph for recommendations by investigating various connection patterns between item entities in KG. Initially, Sun et al. [28] proposed PathSim to compute the path-connected similarity between entities to assess the similarity of connected paths between different entities of the knowledge graph. A Top-N recommendation mutual attention mechanism based on meta-paths was created by MCRec [1] to characterize the contextual representation of users, items, and meta-paths. Using entity and relationship embedding, KPRN [29] builds the extracted path sequence. LSTM is then used to model the extracted path, and a fully connected layer is used to aggregate user preferences on each path. To enhance the user's preference matrix, HeteRec [30] builds several meta-paths for the recommendation system using the knowledge graph and PathSim to extract the preference feature values between users and products. Path-based methods inherently offer justifications for the recommendations made along paths. Unfortunately, their design lacks flexibility when dealing with complicated application circumstances and is heavily reliant on meta-paths, making path selection optimization challenging. Additionally, creating precise pathways calls for labor-intensive work, which requires specialized subject expertise.

The GNN-based method cleverly combines the advantages of embedding-based methods and path-based methods. It integrates multi-hop neighbors into node representation to capture the characteristics and graph structure of nodes, models long-range connections, and integrates the semantic representation of entities and relations with connectivity information. The idea of preference propagation was initially presented by RippleNet [18]. It utilizes the user's past interest set as a seed within the knowledge graph (KG), grows along the links to form multiple ripple sets (Ripple Sets), and then iteratively combines these ripple sets to create the final user representation. Through embedding propagation, KGAT [20] directly aggregates high-order links between users and items by combining interaction and knowledge graphs into heterogeneous graphs. To enhance efficiency, KGIN [31] adds the notion of intent, which discloses the user intent underlying KG interactions and inserts relational embeddings into the aggregate layer. Currently, the most efficient approach is the GNN-based approach since it can identify user preference patterns from KG. However, GNNs have always had a notorious over-smoothing problem, and a large amount of redundant information in KG leads to excessive noise, which has a negative impact on recommendations if it cannot be used reasonably.

2.2. Contrastive Learning

Contrastive learning (CL) has demonstrated promise in computer vision [32] and natural language processing [33] for learning node representations by comparing positive and negative pairs from various perspectives. Through the utilization of self-supervised contrastive learning, several recent studies [34,35] have tackled the issue of data sparsity inherent in recommendation systems.

For the contrastive recommendation method of graph collaborative filtering, to optimize the consistency of multiple views of the same node with other nodes, SGL [34] introduces three operators—node dropout, edge dropout, and random walk—to generate numerous views of a node; LightGCL [36] explores the utilization of global collaboration relationship modeling to enhance the strength of singular values, thereby improving the structure of the user-item interaction graph and achieving unconstrained structure refinement. In the knowledge-aware contrastive recommendation method, MCCLK [37] introduces multi-level cross-view contrastive learning based on the global structure view, local co-operation view, and semantic view to fully extract and integrate additional KG data; KGIC [22] compares the layers of the CF and KG components; KGCL [4] proposes a knowledge graph improvement mode to lessen KG noise in information aggregation and generate a more dependable item knowledge-aware representation. Still, little research

has been carried out to fully realize the enormous potential of contrastive learning in knowledge-graph-based recommendation.

3. Preliminaries

In this section, we begin by introducing the user–item interaction graph and knowledge graph, and then formalize related knowledge-aware recommendation tasks.

In a typical recommendation scenario, there is a user set $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$ and an item set $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$, where u and v represent a single user and item, respectively, and M and N represent the number of users and items. This paper constructs a user–item bipartite graph $\mathcal{G}_u = \{u, y_{uv}, v\}$ based on the user’s historical interaction records with items, where y_{uv} is a binary number. If $y_{uv} = 1$, it represents user u ’s interaction with item v (for example, Purchase, Browse or Click, etc.); otherwise, $y_{uv} = 0$.

In addition to historical interaction data, the knowledge graph (KG) stores a vast amount of real-world structured information related to items in the form of heterogeneous graphs or heterogeneous information networks, such as item attributes and other external common sense [38]. Assume that $\mathcal{G}_k = (h, r, t)$ is a knowledge graph, where $h, t \in \mathcal{E}$ represent the head entity and the tail entity, $r \in \mathcal{R}$ represents the relationship between h and t , and \mathcal{E}, \mathcal{R} represent the entity set and relationship set in the knowledge graph. Here, the entity set \mathcal{E} contains the item set \mathcal{V} and the non-item entity \mathcal{E}/\mathcal{V} . For instance, in the movie recommendation scenario, the triplet (Forrest Gump, Directed by, Robert Zemeckis) signifies that the director of the movie “Forrest Gump” is Robert Zemeckis. It is worth noting that an item $v \in \mathcal{V}$ corresponds to an entity $e \in \mathcal{E}$. For example, the item “Iron Man” in the interaction graph also appears in the knowledge graph as an entity with the same name. This allows this study to establish complex connections between the interaction graph and the knowledge graph through the alignment between items and KG entities, providing additional auxiliary information for interaction data to enhance the modeling of user personalized preferences.

Taking interaction graph $\mathcal{G}_u = \{u, y_{uv}, v\}$ and knowledge graph $\mathcal{G}_k = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$ as input, this paper aims to learn a prediction function $y(u, v)$, whose output indicates the probability of user u interacting with item v .

4. Methodology

In this section, the knowledge contrastive recommendation framework (AKCL-DA) based on a dynamic attention network will be introduced. Figure 1 shows the overall workflow of AKCL-DA.

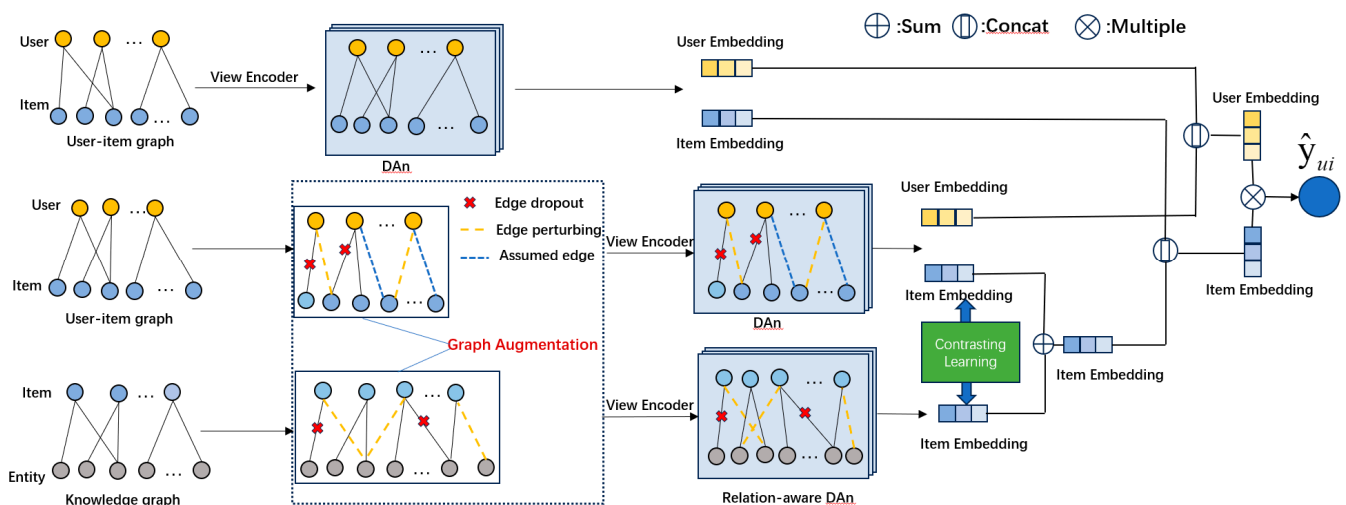


Figure 1. Overall framework of AKCL-DA.

4.1. Adaptive Data Augmentation

Data augmentation is critical to contrastive learning because it allows the model to explore more richly underlying semantic information. With this approach, several views are produced by performing augmentations on the graph (such as the interaction and knowledge graphs). In order to generate interaction and knowledge subgraphs, this study reduces the number of edges on interaction and knowledge graphs in a specific ratio. During this process, considering that important collaborative signals in the interaction graph may be discarded, this paper adds a small number of hypothetical edges to the interaction subgraph. This method can not only reduce the impact of important edge discarding, but also alleviate the problem that users do not observe their favorite items in some cases, resulting in no interaction records. This paper randomly adds hypothesis edges to more comprehensively capture the user's interests and hobbies to better characterize the user's personalized preferences. It should be noted that the knowledge graph itself contains a lot of noise, and this paper does not add hypothetical edges to the knowledge graph. Through edge dropping and adding hypothesis edge operations, this paper obtains the interaction subgraph ($\hat{\mathcal{G}}_u$) and the knowledge subgraph ($\hat{\mathcal{G}}_k$).

Subsequently, a graph augmentation strategy was designed in this study that aims to retain relevant and important edges and remove edges that are irrelevant to recommendations. Simply put, this article introduces a graph augmentation method that first calculates the edge weights w_e^u and w_e^k in $\hat{\mathcal{G}}_u$ and $\hat{\mathcal{G}}_k$, and then calculates the edge sampling probabilities p_e^u and p_e^k based on the edge weights. The Gumbel–Max reparameterization technique is then used [39] after the discrete sampling probabilities are distributed among the continuous variables in $[0, 1]$. In this distribution, the probability value associated with the category having the higher probability tends to be closer to 1, while the probability value associated with the category having the lower probability tends to be closer to 0. For example, edge e obtains the sampling probability p_e by calculating edge weight w_e . If $p_e = 1$, the edge e will be retained; otherwise, it will be discarded.

For the interaction subgraph ($\hat{\mathcal{G}}_u$), first calculate the edge weight according to the following (Formula (1)):

$$w_e^u = MLP_u([e_u^{(0)} \parallel e_v^{(0)}]) \quad (1)$$

where edge $e = (u, v)$, w_e^u represents the importance of the edge, MLP is the abbreviation of multi-layer perceptron, $e_u^{(0)}$ and $e_v^{(0)}$ are the embeddings of users and items, respectively, and the higher the score of w_e^u , the more important the edge is. After obtaining the edge weight w_e^u , this paper uses Formula (2) to calculate the sampling probability p_e^u :

$$p_e^u = \sigma((\log(\epsilon) - \log(1 - \epsilon)) + w_e^u / \tau_u) \quad (2)$$

where the temperature hyper-parameter is used to regulate the approximation, the Sigmoid function is represented by $\sigma(\cdot)$, and the random variable $\epsilon \sim \text{Uniform}(0, 1)$. To aid with end-to-end training, the sampling probability is finally multiplied by the aggregation function as an approximation. Consequently, the interactive subgraph undergoes edge perturbation to produce an interaction-enhanced graph, as follows (Equation (3)):

$$\varphi(\hat{\mathcal{G}}_u) = (\mathcal{V}, M_u \odot \mathcal{E}) \quad (3)$$

where $M_u \in \{0, 1\}$ is the mask vector on the edge set \mathcal{E} . Finally, this paper exploits the GNN message-passing mechanism to obtain user and item embeddings from the interaction-augmented graph.

For the knowledge subgraph ($\hat{\mathcal{G}}_k$), entity information and relationship information must be taken into account simultaneously since it has several various kinds of relationships, each with a very distinct semantic meaning and relevance. Specifically, for a triplet $e = (h, r, t)$, this paper calculates its edge weight w_e^k through Formula (4):

$$w_e^k = MLP_k((e_h^{(0)} \parallel e_r^{(0)} \parallel e_t^{(0)})) \quad (4)$$

where $e_h^{(0)}$, $e_r^{(0)}$ and $e_t^{(0)}$ are the embeddings of head entities, relationships, and tail entities, respectively. Similarly, the higher the w_e^k score, the more relevant the triplet is to the recommendation. Next, this paper calculates the sampling probability p_e^k through Formulas (5) and (6) and uses the same processing method to enhance the knowledge subgraph.

$$p_e^k = \sigma\left(\log(\epsilon) - \log(1 - \epsilon)\right) + w_e^k / \tau_k \tag{5}$$

$$\varphi(\hat{\mathcal{G}}_k) = (\mathcal{V}, M_k \odot \mathcal{E}) \tag{6}$$

where $M_u \in \{0, 1\}$ is the masking vector that determines whether to retain the triplet. Finally, this paper utilizes a graph neural network (GNN) message-passing mechanism to obtain the representation of items from the knowledge-augmented graph.

4.2. Personalized Dynamic Attention Network

GAT (graph attention network) is one of the key models in recommendation systems and graph neural networks, which focuses on effectively aggregating and updating the feature representation of nodes. Traditional GAT has a limitation in that the attention mechanism is static and cannot capture the dynamic contributions of different contextual neighbor nodes to the target node. To address this issue, GATv2 [24] introduced a key modification by swapping the order of applying the linear projection layer and the non-linear function, making the attention score of each node more dynamic and personalized, improving the network’s representation learning ability. This improvement improves the model’s adaptability to changes in graph structure and its ability to resist noise, and this structural enhancement does not incur additional computational costs. Influenced by this, this paper designs a dynamic attention network (DAn) to model the long-range connections between the interaction subgraph and the knowledge subgraph to obtain the optimal representation of the nodes in the graph.

For interaction graphs, we first describe a single layer of (DAn) that first updates node representations and then performs neighborhood aggregation. It is worth noting that each initial node is a free parameter to be trained. In general, the representation vector of an ego node is computed by recursively aggregating and transforming the representations of its multiple neighbors. For node h , to characterize the first-order connectivity of entity h , this paper uses \mathcal{N}_h to represent all nodes connected to h . The linear combination of the self-network of h can be calculated (Equation (7)):

$$e_{\mathcal{N}_h} = \sum_{j \in \mathcal{N}_h} \alpha_{hj} e_j \tag{7}$$

where e_j stands for node j ’s representation, and α_{hj} denotes node h ’s attention score toward node j . The formula for calculating α_{hj} is as follows (Equation (8)):

$$\alpha_{hj} = \frac{\exp(a_u^T \text{LeakyReLU}(W_u e_h + W_u e_j))}{\sum_{k \in \mathcal{N}_h} \exp(a_u^T \text{LeakyReLU}(W_u e_h + W_u e_k))} \tag{8}$$

where a_u and W_u are trainable parameters.

Next, this paper uses the representation of node h and its ego network representation to aggregate into a new representation of h . Specifically, the new representation of node h in layer l is (Calculated by Equation (9)):

$$e_h^{(l)} = \sigma(W_u^{(l)} (e_h^{(l-1)} + e_{\mathcal{N}_h}^{(l-1)})) \tag{9}$$

where $W_u^{(l)}$ is the learnable weight matrix of the l -th layer, and σ is the activation function. Finally, this paper employs an aggregation mechanism [40] to fuse the representations from multiple graph attention network (GAT) layers into a single vector, thereby obtaining the final representation of the user or item.

Although DAN shows powerful modeling capabilities in interaction graph encoding, it does not take relationship types into consideration in modeling, so it is not suitable for encoding knowledge graphs. To put it simply, this paper extends the original attention mechanism, considers the impact of different relationship types on entities, and obtains an encoder relation-aware DAN specifically suitable for heterogeneous graphs. Formally, for a triplet (h, r, t) , the attention score of the head entity h to the tail entity t under the influence of the relationship $r_{\langle h,r \rangle}$ is calculated as follows (Equation (10)):

$$\alpha_{hj} = \frac{\exp(a_k^T \text{LeakyReLU}(W_k e_h + W_r e_{r_{\langle h,t \rangle}} + W_k e_t))}{\sum_{v \in \mathcal{N}_h} \exp(a_k^T \text{LeakyReLU}(W_k e_h + W_r e_{r_{\langle h,t \rangle}} + W_k e_v))} \quad (10)$$

where W_k and W_r are both trainable parameters; e_h , e_t , and $e_{r_{\langle h,t \rangle}}$ are the head entity embedding, tail entity embedding, and the embedding of the relationship $r_{\langle h,t \rangle}$ respectively; $r_{\langle h,t \rangle}$ is the relationship type between h and t . It is worth noting that except for the calculation of attention scores, other modules of knowledge graph encoding are the same as interaction graph encoding.

4.3. Adaptive Contrastive Learning

In this study, a contrastive learning strategy is designed that aims to coherently use collaboration signals and knowledge graph information after acquiring the different embeddings of item v in the interaction graph and the knowledge graph.

Given that the representation spaces of the two views are different, in this study item v 's embedding of e_u^v in the interaction graph and embedding of e_k^v in the knowledge graph are fed into an *MLP* with hidden layers, which are then projected into the same space. Then, we obtain the following (Equations (11) and (12)):

$$z_u^v = \text{MLP}(e_u^v) \quad (11)$$

$$z_k^v = \text{MLP}(e_k^v) \quad (12)$$

Subsequently, this paper treats the same item in the two views as a positive pair (z_u^v, z_k^v) . On the contrary, embedding other items, such as j and v , is regarded as negative samples, such as (z_u^v, z_k^j) , (z_u^j, z_k^v) . A contrastive learning method aims to maximize consistency between positive pairs and minimize consistency between negative pairs, following the definitions of positive pairs and negative pairs provided in this study. Formally, this article adopts the approach from SimCLR [41], and the contrastive loss is defined as follows (Equations (13)):

$$\mathcal{L}_{CL} = -\log \frac{e^{s(z_u^v, z_k^v)/\tau_{cl}}}{\sum_{j \in \mathcal{N} \cup v} e^{s(z_u^v, z_k^j)/\tau_{cl}} + e^{s(z_u^j, z_k^v)/\tau_{cl}}} \quad (13)$$

where τ_{cl} is the temperature hyperparameter, \mathcal{N} is the negative sample set, and $s(\cdot)$ is the cosine similarity of the two vectors.

4.4. Model Prediction

The embeddings of the users and items on the original interaction graph, h_u and h_v , can be obtained by this work after optimization by GNN aggregation and contrastive learning, and the embeddings of users and items on the interaction-augmented graph, namely \hat{h}_u and \hat{h}_u^v . Similarly, this paper obtains the embedding \hat{h}_k^v of the item on the knowledge-augmented graph. \hat{h}_v , which is used for the final prediction, is obtained through the contrastive learning of \hat{h}_u^v and \hat{h}_k^v . The final user and item representations are obtained by concatenating these

embedding representations, and inner products are utilized to predict the matching scores of the resulting representations (Equations (14)):

$$\begin{aligned} e_u &= h_u \parallel \hat{h}_u \\ e_v &= h_v \parallel \hat{h}_v \\ \hat{y}(u, v) &= e_u^T e_v \end{aligned} \tag{14}$$

4.5. Multi-Task Training

This article uses the proposed TransR [26] method to incorporate joint training to further capture the semantic information of entities under various relationships. To effectively capture the semantic linkages between things and relationships throughout the representation learning process, the basic idea is to model entities and relationships through relationship-specific mapping matrices. According to TransR, there are several relationship spaces, and in each relationship space, head and tail entities are represented differently. Specifically, for a given triplet (h, r, t) , its score function can be defined as follows: $f(h, r, t) = \left\| W_r e_h + e_r - W_r e_t \right\|_2^2$, $e_h, e_t \in \mathbb{R}^d, e_r \in \mathbb{R}^k$, representing h, t and r in the triplet (h, r, t) . $W_r \in \mathbb{R}^{k \times d}$ is the relationship space’s transformation matrix; it maps from the d -dimensional entity space to k -dimensional relational space. The lower the score of this function $f(h, r, t) = \left\| W_r e_h + e_r - W_r e_t \right\|_2^2$, the greater the authenticity of the triplet (h, r, t) . The regularization loss \mathcal{L}_{KG} of the knowledge graph is as follows (Equation (15)):

$$\mathcal{L}_{KG} = \sum_{(h,r,t,t') \in \mathcal{G}_k} -\ln \sigma(f(h', r, t')) - f(h, r, t) \tag{15}$$

where h' and t' are entities other than h and t in the knowledge graph, respectively.

This study proposes a multi-task training approach that combines the recommendation task with contrastive learning loss. Specifically, it focuses on reconstructing historical data using the Bayesian Personalized Ranking (BPR) recommendation loss [22], which aims to prioritize higher prediction scores for users’ historical items compared to the unobserved ones:

$$\mathcal{L}_{BPR} = \sum_{(u,i,j) \in \mathcal{O}} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \tag{16}$$

where $\mathcal{O} = \{(u, i, j) \mid (u, i) \in \mathcal{O}^+, (u, j) \in \mathcal{O}^-\}$ represents the training data containing the observed interaction \mathcal{O}^+ and the unobserved interaction \mathcal{O}^+ , and σ is the sigmoid function. In addition, to capture the semantics under different relations in KG and encode KG, this paper adds an auxiliary regularization to KG, and its loss function is \mathcal{L}_{KG} . Finally, this paper utilizes multi-task training techniques to jointly optimize the recommendation loss, contrastive learning loss, and knowledge graph regularization term by minimizing the following objective function, thereby learning the model parameters (Equations (17)):

$$\mathcal{L}_{total} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{CL} + \lambda_2 \mathcal{L}_{KG} \tag{17}$$

where λ_1 and λ_2 are hyperparameters that balance different terms. In practice, λ_1 and λ_2 are fixed to 0.1 and 1, respectively.

5. Experiment

5.1. Experimental Settings

5.1.1. Datasets

To validate the effectiveness of AKCL-DA, this paper utilizes three datasets—Yelp2018, LastFM, and MovieLens—each representing different levels of sparsity. For instance, LastFM exhibits high sparsity, where despite having numerous interactions, the user–item interaction ratio is relatively low compared to its rich set of entities and relations. In contrast, MovieLens has fewer items and interactions, showing a more pronounced sparsity.

These variations in data sparsity provide a diverse experimental setting, allowing for a comprehensive evaluation of the model's performance under different levels of sparsity.

1. Yelp2018: It mostly makes recommendations for merchants, including eateries, pubs, and their locations. Users and items with fewer than ten interactions were removed from the dataset using a 10-core setting.
2. LastFM: These data are collected from the LastFM online music system, containing the listening history of 23,566 users.
3. MovieLens: It is a movie recommendation benchmark dataset containing interaction records from 37,385 users.

For the fairness of the experiment, this article directly uses the dataset partitioning results in KGAT [20]. This study creates a training set and a test set for each dataset by randomly extracting 80% and 20% of each user's interaction history. To update the hyperparameters, this article uses a validation set consisting of 10% randomly selected interactions from the training set. Table 1 displays the fundamental statistical findings for the three datasets.

Table 1. Statistics of experimented datasets.

Stats.	# Users	# Items	# Interactions	# Entities	# Relations	# Triplets
Yelp2018	45,919	45,538	1,183,610	47,472	42	869,603
LastFM	23,566	48,123	3,034,763	106,389	9	464,567
MovieLens	37,385	6182	539,300	24,536	20	237,155

5.1.2. Baselines

This research compares DAKCL with the following models to demonstrate the efficacy of the proposed model.

- BPRMF [5]: It enhances pairwise matrix factorization for implicit feedback by optimizing through Bayesian Personalized Ranking (BPR) loss, enabling the learning of implicit feature representations between users and items.
- FM [21]: Every characteristic is represented as a latent vector, and the inner product of the latent vectors is then calculated to determine the interaction between the features
- NFM [42]: It represents features as latent vectors and learns the interactions between these latent vectors through neural networks.
- CKE [12]: It is an embedding-based method that uses TransR to encode item semantic information and combines collaborative filtering and item knowledge embedding.
- KGCN [10]: It is a GCN-based method, which addresses the high-order dependent context encoding of semantic information in KG.
- LightGCN [7]: By simplifying the graph neural network structure, a lightweight recommendation method based on GCN learns the embedding vectors of users and items.
- KGAT [20]: A propagation-based approach builds a collaborative knowledge graph (CKG) using KG and user-item graphs as building blocks and employs an attention mechanism to generate user and item representations.
- CKAN [43]: A propagation-based method combines collaborative filtering representation space with knowledge graph embedding to further expand KGCN by taking into account the user-item graph and utilizing various neighborhood aggregation algorithms.
- KGIN [31]: A cutting-edge GNN-based method that uses auxiliary item information to try and understand the meaning underlying user-item interactions and gives recommendation interpretability.
- KGCL [44]: A state-of-the-art recommendation system based on contrastive learning proposes a knowledge graph augmentation approach to mitigate KG noise during information aggregation, resulting in a more dependable representation of item knowledge awareness.

5.1.3. Evaluation Metrics

Recall@K and NDCG@K, two commonly used metrics, are employed in this study to completely analyze the performance of this method, with K set to 20. One recall statistic that is utilized to assess recommendation systems is Recall. It calculates the proportion of successfully recommended, genuinely interesting items in a given recommendation list. The normalized discounted cumulative gain of K points is denoted as NDCG@K. It is capable of more precisely assessing the sorting quality performance of the recommendation system by accounting for the relevancy and ordering of suggested items. To compute the metric, this study ranks every item in the training dataset that the user did not interact with. Both approaches have an acceptable range of 0 to 1. The performance of the model recommendation is better the higher the values of the two indicators.

5.1.4. Parameter Settings

This article tunes the critical parameters of each model and implements AKCL-DA and all baseline models in Pytorch (v 1.12.0). To ensure a fair comparison, all models in this study have their embedding dimensions set to 64, their optimizer set to Adam [45], and their embedding parameters initialized using the Xavier method [46]. Every baseline model's parameters are set in accordance with the parameters stated in their paper. Pre-trained MF embeddings are used in this paper to stabilize and expedite model training, in line with earlier research [31]. In this article, the batch size is 4096, the learning rate is 10^{-4} , and the number of GNN layers is fixed at 2; the graph edge discarding rate $\rho = 0.3$. In our AKCL-DA, τ_u and τ_{cl} are tuned at {0.01, 0.05, 0.1, 0.5, 0.7} and {0.5, 0.6, 0.7, 0.8, 0.9}, respectively; τ_k is fixedly set to 0.5; and the proportion of added hypothetical edges in {0.01, 0.02, . . . , 0.05} (hypothetical edge ratio = number of hypothetical edges/total number of edges) is adjusted. This study also employs L2 regularization (with coefficients set to 10^{-3}) and dropout (with a dropout rate of 0.1) at each layer to prevent over-fitting.

5.2. Performance Comparison

In this section, the paper analyzes the performance results of all models. The following observations can be summarized from the results in Table 2:

Table 2. Performance comparison on Yelp2018, LastFM, and MovieLens datasets in terms of Recall@20 and NDCG@20.

Model	Yelp2018		LastFM		Movielens	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
FM	0.0307	0.0206	0.0736	0.0627	0.4175	0.2599
NFM	0.0418	0.0258	0.0673	0.0538	0.3903	0.2357
BPR-MF	0.0499	0.0324	0.0715	0.0618	0.4052	0.2609
CKE	0.0686	0.0431	0.0746	0.0652	0.4106	0.2669
KGCN	0.0532	0.0338	0.0819	0.0705	0.4237	0.2753
LightGCN	0.0682	0.0443	0.0765	0.0686	0.4486	0.3054
KGAT	0.0653	0.0423	0.0877	0.0749	0.4532	0.3007
CKAN	0.0698	0.0441	0.0812	0.0690	0.4314	0.2891
KGIN	0.0712	0.0462	0.0967	0.0847	0.4661	0.3120
KGCL	0.0736	0.0493	0.0899	0.0793	0.4516	0.2967
AKCL-DA	0.0762	0.0518	0.1113	0.0981	0.4742	0.3264
Improv%	3.41%	4.82%	13.12%	13.66%	1.71%	4.41%

In general, AKCL-DA achieves the best performance across all datasets, with improvements of 3.41%, 13.12%, and 1.71% in the Recall@20 metric, respectively. These results verify the superiority of the AKCL-DA method, which is mainly due to the following aspects: (1) By adding an appropriate proportion of hypothetical edges to the interaction graph, we can positively predict items that users may like but have not observed, and also enhance the collaborative signal; (2) The dynamic attention network proposed in this paper

can effectively distinguish the importance of neighbors to improve the embedding of nodes. By aggregating information from the interaction graph and the knowledge graph, it can better capture collaborative signals and semantic feature information; (3) The contrastive view augmentation strategy designed in this paper can identify and keep important and recommendation-related edges unchanged while removing edges that may be unimportant, which can provide high-quality embeddings for contrastive learning.

KGCL, KGIN, CKAN, KGAT, LightGCN, KGCN, and CKE are all graph-based GNN (GAT) models. Evidently, they perform better than BPR-MF, NFM, and FM, which shows that using graph neural networks to aggregate contextual information and model long-range connections can provide better performance for recommendations.

In all datasets, AKCL-DA outperforms FM, NFM, BPR-MF, CKE, KGCN, LightGCN, KGAT, CKAN, and KGCL by 38.89%, 41.04%, 31.50%, 22.85%, 26.17%, 16.99%, 16.62%, 18.65%, and 11.02% on average in NDCG@20 metrics, respectively.

These results illustrate that self-supervised learning provides insights into overcoming data sparsity and cold start by extracting information from unlabeled data. Given the results achieved by KGCL and AKCL-DA, we can find that the item knowledge extracted by contrastive learning can guide the interaction data augmentation mode of self-supervised information. With the help of contrastive learning, the two graph augmentation modes can obtain more robust item knowledge-aware representations; in addition, since contrastive learning only encodes the shared information between the interaction graph and the knowledge graph, it can alleviate the dominance of collaborative signals and suppress noise that is irrelevant to recommendations. However, while AKCL-DA demonstrates significant performance gains, the increased model complexity may lead to higher computational costs, especially in large-scale datasets where efficiency could become an issue. Moreover, when dealing with highly sparse domain-specific data, further optimization of the model's robustness might be necessary. These potential issues will be analyzed in depth in future work, and optimizing the model's performance and efficiency will also be a focus of further research.

5.3. Sensitivity Analysis

In this section, this article will explore the hyperparameter research results of AKCLDA in MovieLens. The trends of Yelp2018 and LastFM are similar to MovieLens, and this article will not go into detail. It is worth noting that this article analyzes one parameter while keeping other vital parameters unchanged.

5.3.1. Effect of Hypothesis Edge Ratio r

This study plots the Recall@20 and NDCG@20 scores to assess the effect of the hypothetical edge ratio on AKCL-DA. The value varies from 0.01 to 0.05. From Figure 2a, we can easily observe that the model achieves the best performance when r is 0.04. Adding hypothetical edges to the interaction graph does improve the recommendation performance to a certain extent, but only an appropriate proportion of hypothetical edges can expand the semantic associations of the knowledge graph and capture user preferences more comprehensively.

5.3.2. Effect of Layer Number L of DAn

Figure 2b shows the Recall@20 and NDCG@20 scores of stacked GNN layers from 1 to 4, and we can observe that the model reaches the best performance when L is 2. As L rises, the model's performance falls, as seen in Figure 2b. This suggests that the model is impacted by the notorious over-smoothing of graph neural networks.

5.3.3. Effect of τ_{cl}

In Figure 2c, this paper plots the effect of the value of τ_{cl} on model Recall@20 and NDCG@20. It is easy to determine that the model performs best at a value of 0.7, indicating

that the performance of the model is especially dependent on determining the proper contrast learning temperature hyperparameter.

5.3.4. Effect of Edge Dropout ρ

In Figure 2d, the effect of the edge dropout rate on the model is shown. This paper changes the dropout rate from 0.5 to 0.9, and the results show that the model reaches the best performance when ρ is 0.3. This demonstrates that excessively high or low edge dropout rates during the construction of interaction and knowledge subgraphs will negatively affect the model's performance. Each round of model training will lose crucial information if the edge dropout rate is too high, whereas each training round will contain more noisy information if the edge dropout rate is too low.

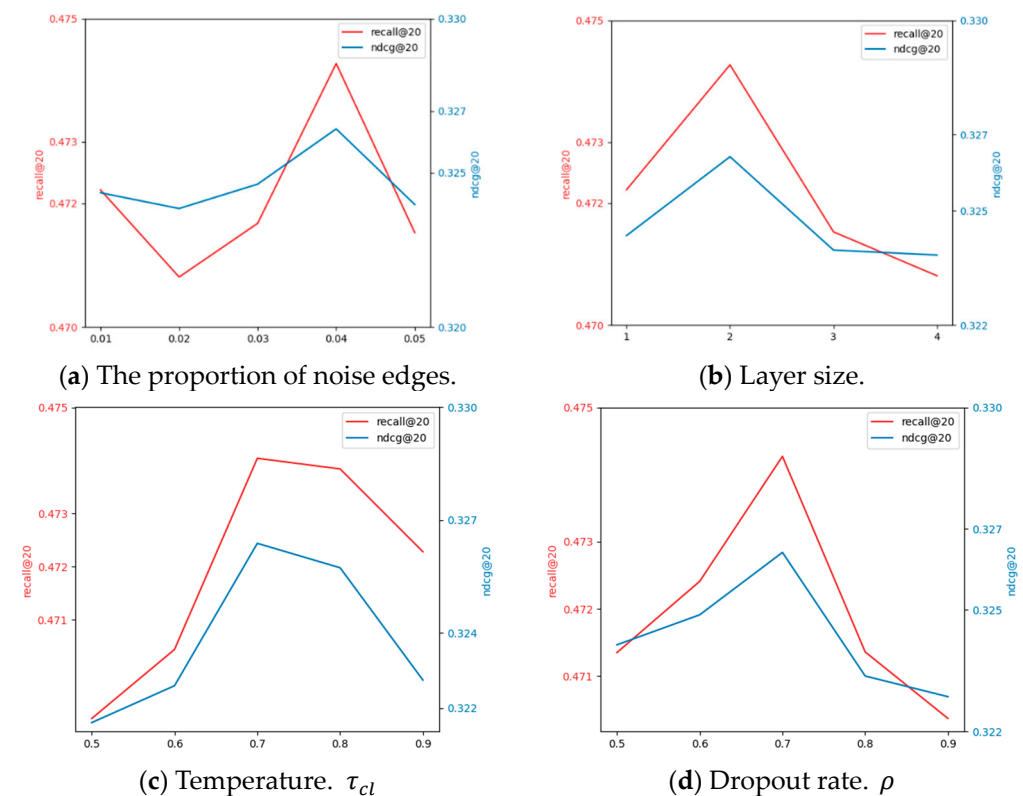


Figure 2. Hyperparameter sensitivity analysis of MovieLens.

5.4. Ablation Studies

To verify the effectiveness of the AKCL-DA component, this paper designs two variants to analyze its performance.

- AKCL-DAw/o Assumed edge: In this variant, we do not add hypothetical edges to the interaction graph and keep other parts consistent with the full model, observing the impact of the results on the entire model.
- AKCL-DAw/o Att: This variant does not use dynamic attention to calculate the importance between nodes but uses the original version of GAT to calculate the attention score.

Table 3 shows the results of the two variants and AKCL-DA. Observing the results reveals the following: (1) Removing hypothetical edges in the interaction graph will affect the performance of the recommendation. Adding hypothetical edges in the interaction subgraph has a certain positive effect on fully capturing user preferences; (2) The dynamic attention network used can effectively capture the relationship between nodes and contribute greatly to improving the embedding of nodes, ensuring high-quality embedding

input for contrastive learning. Ultimately, these results show that by fully leveraging collaborative supervision signals and successfully reducing noise in knowledge graphs, recommendation systems can perform noticeably better.

Table 3. Experimental results for the variants of AKCL-DA.

Model	Yelp2018		LastFM		MovieLens	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
w/o Assumed edge	0.0743	0.0487	0.1103	0.0977	0.4733	0.3257
w/o Att	0.0712	0.0475	0.1092	0.0948	0.4298	0.3228
AKCL-DA	0.0762	0.0518	0.1113	0.0981	0.4742	0.3264

5.5. Benefits of AKCL-DA in Alleviating Knowledge Noise

We evaluate the performance of AKCL-DA on noisy knowledge graphs in this section. To create a large number of entities that are irrelevant to recommendations, 10% noisy triples are to the current knowledge graph while leaving the test set unchanged. Models KGAT, KGCL, and AKCL-DA use knowledge graphs for recommendations after introducing noise. Table 4 presents the study results.

Table 4. Experimental results for mitigating knowledge noise.

Model	Yelp2018		LastFM		MovieLens		Avg. Dec
	Recall	NDCG	Recall	NDCG	Recall	NDCG	
KGAT	0.0623	0.0406	0.0796	0.0692	0.4489	0.2983	4.54%
KGCL	0.0736	0.0472	0.0856	0.0762	0.4503	0.2923	2.89%
AKCL-DA	0.0759	0.0512	0.1109	0.0976	0.4726	0.3257	0.43%

The data in Table 4 demonstrate that, compared with KGAT and KGCL, AKCL-DA has the best average performance in suppressing knowledge graph noise. In general, the method used in this article can effectively capture user preferences and further improve the suppression of knowledge graph noise.

6. Conclusions

This paper focuses on incorporating self-supervised contrastive learning into the recommendation system, using sparse collaborative signals and knowledge graph semantic information naturally. Based on the limitations of existing methods, a new framework AKCL-DA is designed to better achieve dynamic and personalized attention allocation by using a dynamic attention network. Additionally, a graph augmentation strategy is proposed to suppress edge and intra-graph noise irrelevant to the recommendation, to input high-quality graph data for contrastive learning. In addition, to utilize information and guide recommendations in a more balanced way, this study adopts a multi-task learning method to train the model. Extensive experiments on real-world datasets show that AKCL-DA has clear advantages in recommendation performance metrics recall (Recall) and normalized discounted cumulative gain (NDCG). In future research, we will explore other graph augmentation strategies to help the model introduce richer underlying semantic information, such as graph feature transformation, graph structure transformation, and random sampling transformation. In addition, this paper is also interested in exploring more powerful GNNs to more reasonably encode enhanced views. Therefore, future research will continue to explore more effective self-supervised contrastive learning methods to make greater efforts to generate more accurate recommendations.

Author Contributions: Conceptualization, H.L.; Methodology, H.L. and J.Z.; Software, J.Z.; Validation, J.Z.; Resources, B.J.; Writing—original draft preparation, J.Z.; Writing—review and editing, H.Z. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Henan Provincial Science and Technology Project, Project No. 232102210035 and No. 24B520040.

Data Availability Statement: These data were derived from the following resources available in the public domain: <https://www.yelp.com/dataset/download/> (accessed on 22 August 2024), <https://grouplens.org/datasets/hetrec-2011/> (accessed on 22 August 2024) and <https://grouplens.org/datasets/movielens/> (accessed on 22 August 2024).

Acknowledgments: The authors would like to thank the editors and the anonymous reviewers for their helpful comments and suggestions, which have improved the presentation.

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

References

1. Hu, B.; Shi, C.; Zhao, W.X.; Yu, P.S. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1531–1540.
2. Wang, C.; Zhu, H.; Zhu, C.; Qin, C.; Xiong, H. Setrank: A setwise bayesian approach for collaborative ranking from implicit feedback. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 6127–6136.
3. Wang, C.D.; Deng, Z.H.; Lai, J.H.; Philip, S.Y. Serendipitous recommendation in e-commerce using innovator-based collaborative filtering. *IEEE Trans. Cybern.* **2018**, *49*, 2678–2692. [[CrossRef](#)] [[PubMed](#)]
4. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
5. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
6. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [[CrossRef](#)]
7. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25–30 July 2020; pp. 639–648.
8. Pan, W.; Wei, W.; Mao, X.L. Context-aware entity typing in knowledge graphs. *arXiv* **2021**, arXiv:2109.07990.
9. Li, S.; Jia, Y.; Wu, Y.; Wei, N.; Zhang, L.; Guo, J. Knowledge-Aware Graph Self-Supervised Learning for Recommendation. *Electronics* **2023**, *12*, 4869. [[CrossRef](#)]
10. Wang, H.; Zhao, M.; Xie, X.; Li, W.; Guo, M. Knowledge graph convolutional networks for recommender systems. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3307–3313.
11. Wang, H.; Zhang, F.; Xie, X.; Guo, M. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1835–1844.
12. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.Y. Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 353–362.
13. Cao, Y.; Wang, X.; He, X.; Hu, Z.; Chua, T.S. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 151–161.
14. Xie, F.; Zhang, Y.; Przystupa, K.; Kochan, O. A Knowledge Graph Embedding Based Service Recommendation Method for Service-Based System Development. *Electronics* **2023**, *12*, 2935. [[CrossRef](#)]
15. Wang, H.; Zhang, F.; Hou, M.; Xie, X.; Guo, M.; Liu, Q. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining 2018, Los Angeles, CA, USA, 5–9 February 2018; pp. 592–600.
16. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-Relational data. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013. Available online: <https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html> (accessed on 21 August 2024).
17. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
18. Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 417–426.
19. Zhao, H.; Yao, Q.; Li, J.; Song, Y.; Lee, D.L. Meta-graph based recommendation fusion over heterogeneous information networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 635–644.

20. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
21. Rendle, S.; Gantner, Z.; Freudenthaler, C.; Schmidt-Thieme, L. Fast context-aware recommendations with factorization machines. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 25–29 July 2011; pp. 635–644.
22. Zou, D.; Wei, W.; Wang, Z.; Mao, X.L.; Zhu, F.; Fang, R.; Chen, D. Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 2817–2826.
23. Jing, M.; Zhu, Y.; Zang, T.; Wang, K. Contrastive self-supervised learning in recommender systems: A survey. *ACM Trans. Inf. Syst.* **2023**, *42*, 1–39. [[CrossRef](#)]
24. Brody, S.; Alon, U.; Yahav, E. How attentive are graph attention networks? *arXiv* **2021**, arXiv:2105.14491.
25. Yang, B.; Yih, W.T.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv* **2014**, arXiv:1412.6575.
26. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
27. Rakhlin, A. Convolutional neural networks for sentence classification. *GitHub* **2016**, *6*, 25.
28. Yu, X.; Ren, X.; Gu, Q.; Sun, Y.; Han, J. Collaborative filtering with entity similarity regularization in heterogeneous information networks. *IJCAI HINA* **2013**, *27*, 1–6.
29. Wang, X.; Wang, D.; Xu, C.; He, X.; Cao, Y.; Chua, T.S. Explainable reasoning over knowledge graphs for recommendation. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5329–5336.
30. Yu, X.; Ren, X.; Sun, Y.; Sturt, B.; Khandelwal, U.; Gu, Q.; Norick, B.; Han, J. Recommendation in heterogeneous information networks with implicit user feedback. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 347–350.
31. Wang, X.; Huang, T.; Wang, D.; Yuan, Y.; Liu, Z.; He, X.; Chua, T.S. Learning intents behind interactions with knowledge graph for recommendation. In *Web Conference*; ACM: New York, NY, USA, 2021; pp. 878–887.
32. Li, Y.; Zhang, Y.; Gao, Y.; Xu, B.; Liu, X. FCL: Pedestrian Re-Identification Algorithm Based on Feature Fusion Contrastive Learning. *Electronics* **2024**, *13*, 2368. [[CrossRef](#)]
33. Fu, H.; Zhou, S.; Yang, Q.; Tang, J.; Liu, G.; Liu, K.; Li, X. LRC-BERT: Latent-representation contrastive knowledge distillation for natural language understanding. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, Vancouver Convention Centre, VA, Canada, 2–9 February 2021; Volume 35, pp. 12830–12838.
34. Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; Xie, X. Self-supervised graph learning for recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, 11–15 July 2021; pp. 726–735.
35. Wang, L.; Jin, D. A Time-Sensitive Graph Neural Network for Session-Based New Item Recommendation. *Electronics* **2024**, *13*, 223. [[CrossRef](#)]
36. Cai, X.; Huang, C.; Xia, L.; Ren, X. LightGCL: Simple yet effective graph contrastive learning for recommendation. *arXiv* **2023**, arXiv:2302.08191.
37. Zou, D.; Wei, W.; Mao, X.L.; Wang, Z.; Qiu, M.; Zhu, F.; Cao, X. Multi-level cross-view contrastive learning for knowledge-aware recommender system. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1358–1368.
38. Gao, B.; Liu, T.Y.; Wei, W.; Wang, T.; Li, H. Semi-supervised ranking on very large graphs with rich metadata. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 96–104.
39. Luo, D.; Cheng, W.; Xu, D.; Yu, W.; Zong, B.; Chen, H.; Zhang, X. Parameterized explainer for graph neural network. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 19620–19631.
40. Liu, Y.; Xu, C.; Chen, L.; Yan, M.; Zhao, W.; Guan, Z. TABLE: Time-aware Balanced Multi-view Learning for stock ranking. In *Knowledge-Based Systems*; Elsevier: Amsterdam, The Netherlands, 2024; p. 112424.
41. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, 13–18 July 2020; pp. 1597–1607.
42. He, X.; Chua, T.S. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 355–364.
43. Wang, Z.; Lin, G.; Tan, H.; Chen, Q.; Liu, X. CKAN: Collaborative knowledge-aware attentive network for recommender systems. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25–30 July 2020; pp. 219–228.
44. Yang, Y.; Huang, C.; Xia, L.; Li, C. Knowledge graph contrastive learning for recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1434–1443.

-
45. Kingma, D.P. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
 46. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010; pp. 249–256.

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