JCSE Journal of Computing Science and Engineering

Regular Paper
Journal of Computing Science and Engineering,
Vol. 18, No. 3, September 2024, pp. 169-180

KHG-Aclair: Knowledge Hypergraph-Based Attention with Contrastive Learning for Recommendations

Hyejin Park, Taeyoon Lee, and Kyungwon Kim*
Korea Electronics Technology Institute, Seoul, Korea
hyejinpark@keti.re.kr, tylee814@keti.re.kr, kwkim@keti.re.kr

Abstract
Information overload and complex user interactions make it difficult to retrieve valuable data. Recommendation systems have become crucial in addressing this challenge by providing users with relevant information and items. Collaborative filtering-based recommendation methods, which are commonly used in this context, often suffer from data scarcity, thus limiting their effectiveness for users with insufficient interaction data. To overcome this problem, knowledge graphs have been integrated into recommendation systems to enhance user and item representation through the implementation of additional semantic relatedness. Despite their potential utility, most recommendation models assume binary relations within knowledge graphs, thereby overlooking the high-order relationships that are prevalent in knowledge graphs. Knowledge hypergraphs, which can capture complex and multi-dimensional relationships, offer a solution to this limitation. This paper proposes KHG-Aclair, a novel recommendation system that leverages hypergraphs to uncover hidden features within knowledge graphs, thus enhancing recommendation accuracy and insight. We have transformed the Freebase knowledge graph into a knowledge hypergraph and made this dataset publicly available. KHG-Aclair also incorporates contrastive learning to refine the knowledge hypergraph, thus reducing noise and improving representation for less popular items. Altogether, our model demonstrates strong generalizability, as it achieves high performance across multiple datasets, thus indicating that it can serve as a versatile solution for various recommendation systems. Our implementation codes are available at https://github.com/HBD-NGC1316/KHG-Aclair.

Category: Smart and Intelligent Computing

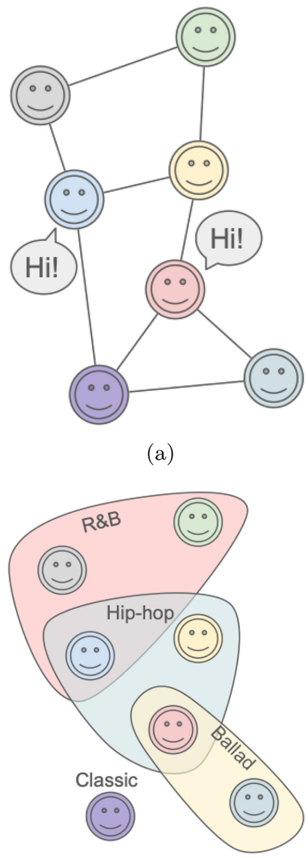
Keywords: Recommendation system; Self-supervised learning; Contrastive learning; Knowledge hypergraph; Knowledge graph

I. INTRODUCTION

The continued advancement of the internet has recently led to the growth of massive online platforms, resulting in information overload and an explosion of data. This has made it difficult to search for valuable information due to the increasingly complex interactions between users. Consequently, recommendation systems have become increasingly important by providing users with relevant information and items. Among the various types of recommendation methods, collaborative filtering is commonly used to enhance recommendation accuracy. However, collaborative filtering-based methods often suffer from data scarcity, which limits their effectiveness for users with insufficient interaction data, and the lack of interaction data hinders the basis for effective recommendations [1-

Open Access http://dx.doi.org/10.5626/JCSE.2024.18.3.169 http://jcse.kiise.org

Received 1 July 2024; **Accepted** 10 September 2024
*Corresponding Author



(a)

(b)

Fig. 1. Comparison between conventional graphs and hypergraphs. (a) Conventional graphs represent relationships as single edges between pairs of entities. (b) Hypergraphs introduce hyperedges that can connect multiple entities simultaneously, thus capturing higher-dimensional relationships and eventually revealing deeper insights within knowledge graphs.

3]. To overcome this problem, knowledge graphs have been increasingly incorporated into recommendation systems to enhance user and item representation by encoding additional item-wise semantic relatedness [4-6].

Most recommendation models assume that all relations in the knowledge graph are binary [7], which limits their ability to utilize high-order relationships. Observations from publicly accessible knowledge graphs like Freebase indicate that over 61% of relations exceed binary connections [8]. Therefore, to better understand the structural characteristics of knowledge graphs, it is essential to consider high-order relationships. While conventional graphs only identify relationships connecting two entities via a single edge, hypergraphs define new high-order relationships through hyperedges, thus enhancing our understanding of the hidden features within knowledge graphs. For example, as can be seen in Fig. 1(a), a conventional graph depicts a structure where each node is simply connected, such as friend relationships in a social network. By contrast, a hypergraph, as shown in Fig. 1(b), reveals different relationships that are not found in simple friend connections. In the context of musical taste, these hyperedges can represent music genres like R&B, hip-hop, etc., thus allowing us to discover information about individual relationships from the perspective of personal music preferences that are hidden within the simple social network. Hypergraphs express complex and diverse relationships among nodes, eventually uncovering new relationships not visible in conventional graphs. The ability to reveal hidden relationships in this way is a key feature of hypergraphs.

Simultaneously, the performance of recommendation models utilizing knowledge graphs depends heavily on the refinement of the knowledge graph. When popular nodes are connected to numerous edges, less popular items may not receive attention, in turn compromising the recommendation system's performance. Moreover, irrelevant information related to items or entities in the neighborhood nodes of the knowledge graph can trigger significant noise, further deteriorating performance. These challenges can be alleviated by employing contrastive learning [9], which trains the model to differentiate between similar and contrasting pairs.

Therefore, we propose KHG-Aclair, a knowledge hypergraph-based attention with contrastive learning for recommendation systems. KHG-Aclair provides information by considering complex relationships beyond traditional binary relations. Hypergraphs can uncover complex and high-order relationships between nodes that are not visible in conventional graphs. This new methodology that is based on attention and contrastive learning offers better insights, thus allowing for the delivery of more accurate and useful user recommendations. It is difficult to handle the complexity of hypergraph data. To address this, we have developed an algorithm that can be used to construct a hypergraph from the Freebase dataset. We have transformed the publicly available knowledge graph, Freebase, into a knowledge hypergraph to enhance the performance of the recommendation system, and we have made this dataset public. This strategic measure is expected to help improve the efficiency of our recommendation system while also fostering diversity in the recommendation system.

The contributions of this paper can be summarized as follows:

- We propose KHG-Aclair, a novel recommendation system that leverages hypergraphs to uncover high-order relationships, ultimately providing more accurate and insightful recommendations than conventional graph-based methods.
- Our proposed model incorporates contrastive learning to effectively refine the knowledge hypergraph,

mitigating noise and improving the representation of less popular items, thereby enhancing overall recommendation performance.
- We transform the publicly available Freebase knowledge graph into a knowledge hypergraph and make this enriched dataset publicly available, to serve as a resource for further exploration and development.

Our model demonstrates strong generalizability, as it achieves high performance across multiple datasets, making it a versatile solution that is applicable to a wide range of recommendation systems.

The rest of this paper is organized as follows: Section II introduces and analyzes related research in this area. Section III defines the fundamental concepts used in this study. Section IV provides a detailed description of our method. Sections V and VI outline the experimental settings and discuss our model's performance. Section VII explores future research directions. Finally, Section VIII concludes this paper.

II. RELATED WORK

A. Knowledge Graph-Enhanced Recommendation

Prior research on knowledge graph-enhanced recommendation systems, such as knowledge graph attention network (KGAT) [5], knowledge graph convolutional network (KGCN) [10], and knowledge graph-based intent network (KGIN) [11], has achieved enhanced rich semantics by leveraging the relationships between entities. KGAT recursively propagates embeddings from entities' neighbors to refine the embeddings, and it also applies an attention mechanism. KGCN samples neighbors for each user and item as a receptive field to capture users' potential long-term interests. KGIN further considers user latent factors affecting various relationships in the knowledge graph and injects relational embeddings into the aggregation layer. Although these studies consider various relations and capture long-term interests, they do not account for hyper-relations, which makes it difficult to capture high-order relations.

B. Contrastive Learning for Recommender System

Within recommendation systems, self-supervised learning (SSL) has emerged as a new trend for addressing the data sparsity problem. Among SSL methods, contrastive learning acquires mutual information between two representations by learning positive and negative samples. Neighborhood-enriched contrastive learning (NCL) [12] attempts to incorporate contrastive learning into the modeling of user-item interactions for collaborative filtering. Self-supervised graph learning (SGL) [13] analyzes the side effect of contrastive learning on a user-item graph, which involves mining hard negative examples. It employs strategies such as node dropout, edge dropout, and random walks on graph connections. Multi-modal self-supervised learning (MMSSL) [14] focuses on modeling user interaction patterns across different modalities, and the authors of that study propose cross-modal contrastive learning. Knowledge graph contrastive learning (KGCL) [9] suggests that graph-based contrastive learning be used in knowledge graphs to alleviate noise and long-tail problems. This approach explores knowledge graph semantics, thus enabling knowledge-guided recommendations in binary relations.

III. PRELIMINARIES

In this section, we introduce the definition of the knowledge hypergraph and define the recommendation task.

A. Knowledge Hypergraph

A *knowledge hypergraph* is made up of a finite set of entities $\mathcal{E}$, a finite set of relations $\mathcal{R}$, and a collection of tuples $\mathcal{T}$. Each tuple $\tau \in \mathcal{T}$ can be expressed as $r(z_1, z_2, ..., z_k)$, where $r \in \mathcal{R}$, and each $z \in \mathcal{E}$. The number of elements in each tuple is called the arity $|r|$. A world defines the truth that all tuples in $\mathcal{T}$ are true, while those not in $\mathcal{T}$ are false. A knowledge hypergraph is formed from a subset of the tuples $\tau_0 \subseteq \tau$ [7].

A *knowledge graph*, on the other hand, consists of a finite set of entities $\mathcal{E}'$, a finite set of relations $\mathcal{R}'$, and a collection of triples $\mathcal{T}'$. Each triple τ' in $\mathcal{T}'$ is expressed as $r'(z_i, z_t)$, where r' is a relation from $\mathcal{R}'$, and each z_i is an entity from $\mathcal{E}'$. The key difference here is that, in a knowledge graph, all relations are binary ($|r'| = 2$), while in a knowledge hypergraph, relations can have any arity, thus allowing for more complex relationships.

A *hypergraph* extends the concept of a traditional graph by including hyperedges, which can link more than two nodes [15, 16]. Formally, a hypergraph H is defined as $H = (V, E)$, where V is the set of nodes denoted as $V = \{v_1, ..., v_i, ..., v_n\}$ and E is the set of edges and hyperedges. The hyperedges are denoted as $E = \{e_1, ..., e_i, ..., e_m\}$, with each hyperedge e_i connecting two or more nodes. Because of this, a hypergraph can represent complex relationships between nodes more effectively than a conventional graph.

The structure of a hypergraph H can also be described using an incidence matrix $A \in \mathbb{R}^{n \times m}$, where the entries are defined as follows:

$$A_{i,j} = \begin{cases} 1 & \text{if } v_i \in e_j, \\ 0 & \text{if } v_i \notin e_j. \end{cases} \quad (1)$$

In a general scenario, each node in a hypergraph may have a d-dimensional attribute vector. Consequently, the attributes of all nodes can be represented as $X = [x_1, x_2, ..., x_n]^{\top} \in \mathbb{R}^{n\times d}$; for simplicity, the entire hypergraph can be denoted as $H = (A, X)$.

B. Problem Definition

Consider the user set $\mathcal{U} = \{u_1, u_2, ..., u_p\}$ and the item set $\mathcal{I} = \{i_1, i_2, ..., i_q\}$. The items that a user u has consumed are represented by $\mathcal{C}$. We use a binary matrix $R \in \mathbb{R}^{p\times q}$ to store the interactions between users and items. In matrix $\mathcal{C}$, $c_{u,i} = 1$ signifies that user u has consumed item i, while $c_{u,i} = 0$ indicates that user u either has not been exposed to item i or is not interested in it. This paper's focus is on generating top-K recommendations.

IV. METHODOLOGY

A. Item Knowledge Hypergraph Construction

In this section, we describe how an item knowledge hypergraph can be constructed from the Freebase knowledge graph (Fig. 2). The item knowledge hypergraph is a knowledge hypergraph that contains external knowledge about items, and the process of constructing it is outlined in Algorithm 1. We only deal with a 1-hop relation and nodes related to items from the knowledge graph. While expanding the search scope might yield more information, it also introduces too much noise in the process. The method used to search a 1-hop relation and nodes related to items follows the approach in [17].

Algorithm 1 takes as input the item set $\mathcal{I}$, the set of user-item interaction pairs $\mathcal{C}$, and the entity set $\mathcal{E}'$ consisting of pairs of items and their corresponding knowledge graph entities. We also obtain as input the set $\mathcal{T}'$ of triples consisting of 1-hop nodes and relations searched from entities that are associated with the items i that user u interacts with. The output of Algorithm 1 is the knowledge hypergraph matrix A.

Lines 1-2 extract the common item set $\mathcal{I}'$ from the user-item interaction set $\mathcal{C}$, item-entity pair set $\mathcal{E}'$, and knowledge graph triple $\mathcal{T}'$ to generate the knowledge hypergraph for these common items. Here, set $\mathcal{T}'''$ consists of triples of the common item entity $\tau_{i'}$, the relation τ_r, and the tail entity τ_z, where $\tau_{i'}$ is an entity corresponding to the item, τ_r is the relation, and τ_z is a tail entity connected to the opposite side of the relation. Lines 3-6 insert tuples of set $\mathcal{T}''$ for the common items into a dictionary D to store the triples of the knowledge graph related to the items. Lines 7-12 utilize the relations of the triples associated with items, specifically using them as indices to store nodes at the corresponding positions. Lines 13-21 construct the entire matrix of the item knowledge hypergraph. For each hyperedge τ_r stored in list L, nodes $\tau_{i'}$ and τ_z associated with τ_r are used as indices to insert a 1 into the initialized two-dimensional matrix A having 0s. Finally, in line 22, the generated adjacency matrix of the knowledge hypergraph A is returned. We also construct item knowledge hypergraphs for three recommendation system evaluation datasets. The statistics for these datasets are presented in Table 1.

B. Hypergraph Attention Layer

Existing knowledge graph-based recommendation models often assume that all relations in the knowledge graph are

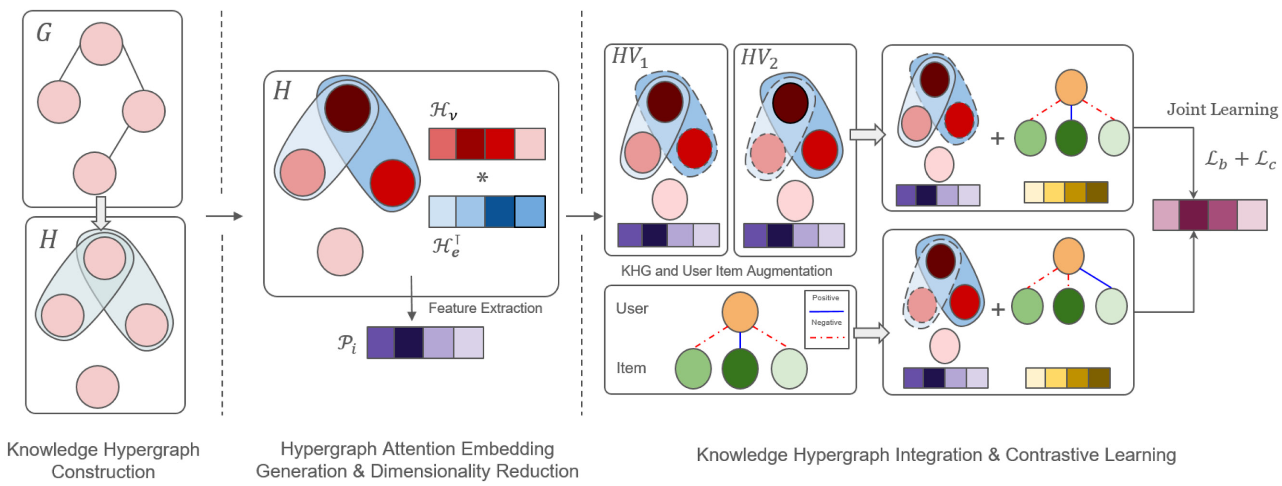


Fig. 2. The structure of our proposed model, KHG-Aclair, is as follows: we first convert the knowledge graph into a hypergraph and extract features through an attention layer. We then optimize these extracted features during the learning process by combining them with user and item information using contrastive learning.

Algorithm 1 Item-Knowledge Hypergraph Construction

Input: $\mathcal{I} = \{i_k \mid 1 \le k \le q, k \in \mathbb{N}\}$, ▷ The entire set of items
$C_{u,i} = \{(u,i) \mid u \in \mathcal{U}, i \in \mathcal{I}\}$, ▷ User-item interaction pair
$\mathcal{E}'_{i,z'} = \{(i_k, z_k) \mid \text{Entity } z \text{ corresponds to item } i\}$, ▷ Item-entity pairs
$\mathcal{T}'_{z',r,z} = \{(z_i, r_j, z_k) \mid z \in \mathcal{E}', r \in \mathcal{R}'\}$ ▷ Knowledge graph triples
Output: Knowledge Hypergraph Adjacency Matrix $a_{i,j}$

$\mathcal{I}' = C_{u,i_j} \cap \mathcal{E}'_{i,z'} \cap \mathcal{T}'_{z',r,z}$ ▷ Common item set $\mathcal{I}' = \{i'_1, i'_2, ..., i'_y\}$
$\mathcal{T}'' = \{\tau \mid \tau = (\tau_{i'}, \tau_r, \tau_z)\}$ ▷ Set containing tuples $(\tau_{i'}, \tau_r, \tau_z)$ as elements
Item triple dictionary D
for `each item` i' `in` $\mathcal{I}'$ **do**
  $D[i'] \leftarrow (\tau_{i'}, \tau_r, \tau_z)$ ▷ Insert tuple τ corresponding to Item i'
end for
Item hyperedge list L
for `each item` i' `in` D **do**
  for `each relation` τ_r `in` $D[i']$ **do**
    $L[i'][\tau_r] \leftarrow \tau_{i'}, \tau_z$ ▷ Insert τ_z based on τ_r associated with items
  end for
end for
Initialize the hypergraph adjacency matrix A to all zeros.
for `each item` i' `in` L **do**
  for `each relation` τ_r `in` $L[i']$ **do**
    for `each nodes` $\tau_{i'}, \tau_z$ `in` $L[i'][\tau_r]$ **do**
      $A[\tau_r][\tau_z] \leftarrow 1$ ▷ Set relation between τ_r and τ_t
      $A[\tau_r][\tau_{i'}] \leftarrow 1$ ▷ Set relation between τ_r and $\tau_{i'}$
    end for
  end for
end for
return A

Table 1. Experiment statistics for the KHG-Aclair dataset

	MovieLens	Last FM	Amazon-Book
Number of users	943	971	11,504
Number of items	1,405	4,913	3,852
Sparsity	0.0668	0.00639	0.00080
Knowledge hypergraph			
Hyper edges	470	145	312
Nodes	392,849	7,352,735	94,374

binary. However, transforming high-order relationships into binary relations often fails to yield satisfactory results, thus necessitating the development of a knowledge hypergraph-based recommendation system model.

Our model is inspired by hypergraph graph attention (HyperGAT) [18], which is implemented using a dual-attention mechanism. This mechanism not only captures interactions among high-order relationships during the learning process of node representations but also allows for other granular details to be emphasized. Moreover, the layers of HyperGAT implement two functions: one for representing the features of nodes and another for representing the features of hyperedges (Eqs. 2, 3) as follows:

$$f_v(e) = \mathrm{AGGR}_v(\{v \in V_e\}) \tag{2}$$

The function f_v aggregates the features of node v to the hyperedge e. V_e represents the set of nodes connected to the hyperedge e, and AGGR_v is the function that aggregates the features of nodes into the hyperedge.

$$f_e(v) = \mathrm{AGGR}_e(\{e \in E_v\}) \tag{3}$$

$f_e(v)$ represents the features of node v, which are obtained by aggregating the features of hyperedge e connected to v. E_v denotes the set of hyperedges connected to node v, while AGGR_e is the function that aggregates the features of hyperedges into nodes.

C. Hypergraph Embedding Generation and Dimensionality Reduction

In this section, we describe the process by which node embeddings associated with items are generated based on graph embeddings that are produced by the hypergraph attention layer (HGAT). We also explain the method of dimensionality reduction of the embeddings using principal component analysis (PCA) [19]. The graph embeddings generated through the HGAT consider the features of each node, incorporating information within the graph structure. These embeddings are defined by the following Eq. (4):

$$\mathcal{H} = \mathrm{HGAT}(\mathcal{H}_{\mathrm{v}}, \mathcal{H}_{\mathrm{e}}^{\top}) \tag{4}$$

Here, $\mathcal{H}_{\mathrm{v}}$ is the embedding matrix representing the features of the nodes, $\mathcal{H}_{\mathrm{e}}^{\top}$ is the transposed edge embedding matrix, and $\mathcal{H} \in \mathbb{R}^{n \times d}$ denotes the embedding matrix obtained through the graph attention layer. Next, for each item i, the embeddings are extracted using the indices of the associated nodes. This is expressed by Eq. (5) as follows:

$$\mathcal{P}_i = \mathrm{CONCAT}(\mathcal{H}[v_1], \mathcal{H}[v_2], \ldots, \mathcal{H}[v_k]) \tag{5}$$

Here, $v_1, v_2, ..., v_k$ represent the indices of the nodes associated with item i. By extracting the embeddings of the nodes associated with each item, $\mathcal{P} \in \mathbb{R}^{k \times d}$, we can represent the characteristics of the item within the graph structure. Then, for each item, the embeddings are reconstructed into a two-dimensional form, and dimensionality reduction is performed using PCA. This can be expressed using the following Eq. (6):

$$(\mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Phi}^{\top}) = \mathrm{PCA}(\mathrm{Reshape}(\mathcal{P}_i), \mathrm{s} = 200) \tag{6}$$

Here, $\mathbf{\Pi} \in \mathbb{R}^{s \times d}$ represents the reduced embeddings obtained through PCA, $\mathbf{\Sigma}$ is a diagonal matrix with singular values as its diagonal elements, and $\mathbf{\Phi}$ represents the matrix of eigenvectors obtained through PCA. $\mathbf{s}$ denotes the number of dimensions to which the embeddings are reduced. By defining the final dimensionally reduced embeddings as $\hat{\mathcal{P}} = \mathbf{\Pi}$, we can effectively summarize the information of nodes associated with items and ultimately represent the embedding space more efficiently through dimensionality reduction.

D. Knowledge Hypergraph Integration and Contrastive Learning

Contrastive learning serves as a framework for integrating knowledge hypergraphs and enhancing user and item representations. Contrastive learning can be used to handle various relationships within knowledge hypergraphs, expand data, suppress noise, and leverage external signals to improve model flexibility. Our contrastive learning framework draws inspiration from KGCL [9].

According to the proposals made by KGCL, we combine KHG embeddings and contrastive learning to evaluate the consistency of items and identify items that are less sensitive to structural changes. By leveraging item consistency scores, we augment the user-item interaction graph and learn the representations of items and users through co-contrastive learning. In this process, while following the prior ranking model [9, 20], the KHG-Aclair leverages the Bayesian personalized ranking (BPR) recommendation loss $\mathcal{L}_{\mathrm{b}}$. Following the prior study [9], we adopt a contrastive loss $\mathcal{L}_{\mathrm{c}}$ based on the InfoNCE [21] loss in our model. We improve the accuracy and robustness of the recommendation system by combining BPR loss and contrastive loss.

The overall loss function to train our contrastive learning model can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_{\mathbf{b}} + \lambda_1 \mathcal{L}_c + \lambda_2 \|\theta\|_2^2. \tag{7}$$

where λ_1 and λ_2 are hyperparameters that are used to control the weights of self-supervised signals and regularization terms. θ represents the learnable model parameters.

V. EXPERIMENTS

A. Experimental Setup

All models and algorithms are implemented using Python 3.8 and PyTorch 1.9. We conduct the experiments on an Ubuntu 20.04 LTS System server with a 48-core CPU, 512 GB RAM, and an NVIDIA RTX A5000 GPU. The experiments were repeated 10 times, and the best result was selected. We optimize our model with the Adam optimizer and fix the embedding size at 200. To find the optimal set of hyperparameters for KHG-Aclair, we use Bayesian hyperparameter optimization Optuna [22], which is designed to accelerate the tuning process for each dataset. Specifically, the learning rate is tuned within the range of [1e-5, 1e-2], while the batch size and the dropout rate are searched within the ranges of [512, 4096] and [0.0, 0.5], respectively. We select the contrastive loss balance parameter λ_1 from the range [0.0, 1.0].

B. Datasets

To achieve a diverse and representative evaluation, we conducted experiments on three different real-world datasets: Amazon-Book for book recommendations, Last FM for music recommendations, and MovieLens 1M for movie recommendations. The Amazon-Book dataset is a dataset that is widely utilized for recommendation systems; we specifically focused on a subset covering timestamps from January 2014 to December 2014. The Last FM dataset used in our study was released by KGAT [5]. The MovieLens 1M dataset includes user IDs, movie IDs, and timestamps. After preprocessing the three datasets, we mapped their items to Freebase entities. Freebase [9, 17] is a Knowledge Base Dataset that is maintained by Google Inc., and which stores triples in the form of <`head`, `relation`, `tail`>. In constructing the knowledge hypergraph, we collected entities within one hop, as modeling two or more hops often leads to excessive computational time and noise. To mitigate noise issues, we also filter out KHG entities involved in less than 10 relations. We used 80% of the ratings from each dataset for training and the remaining 20% for testing. Table 1 summarizes the statistics of user-item interactions and knowledge hypergraphs for the three evaluation datasets.

C. Baselines

To demonstrate the effectiveness of our approach, we compare the performance of our proposed KHG-Aclair with those of knowledge graph-based recommendation methods. The hyperparameter settings for the baselines were tuned according to the papers in which they were originally described. The methods are briefly introduced as follows:

KGCL [9]: This aims to provide graph contrastive learning for knowledge graphs to reduce potential knowledge noise. The knowledge graph contrastive learning signals are introduced to train unbiased user-item interactions.

KGRec [23]: This model is a state-of-the-art method based on knowledge guidance in recommendation, which improves performance by using a masking and reconstruction module to emphasize rational knowledge. This knowledge is further used to facilitate knowledge-aware cross-view contrastive learning

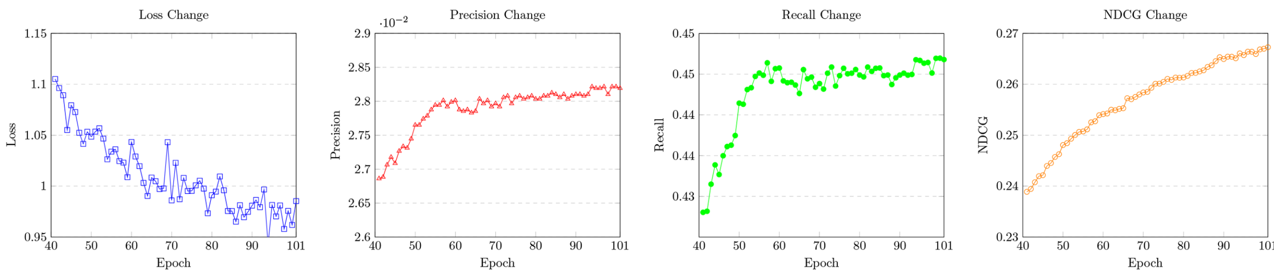
VI. RESULTS AND ANALYSIS

A. Evaluation Metrics

To evaluate the performance of our model, we used three major metrics adopted from other studies: Recall@N, Precision@N, and NDCG@N [5, 9, 23]. Each metric makes its evaluations based on the average results for all users in the test set. Here, N is set to 20 by default. Recall@N measures the proportion of relevant items among the top N items recommended by the model, thus indicating how many relevant items the model successfully identified. Precision@N indicates the proportion of relevant items among the top N items recommended, thus assessing the accuracy of the model's recommendations. Lastly, NDCG@N measures the validity of the order of recommended items, with higher weights given to items that are ranked higher, thus evaluating the overall quality of the recommendations. These metrics allow us to evaluate the model's performance from multiple perspectives, ultimately contributing to more objective comparisons of our model's performance with that of other studies.

B. Overall Performance

In this section, we evaluate the performance of our proposed model, KHG-Aclair, in comparison to the baseline models KGCL and KGRec when used with each of three datasets: Amazon-Book, Last FM, and MovieLens (Figs. 3–5).



(a) (b) (c) (d)

Fig. 3. Training loss and evaluation metrics evolution over epochs on Amazon-Book dataset: (a) loss, (b) precision, (c) recall, and (d) NDCG.

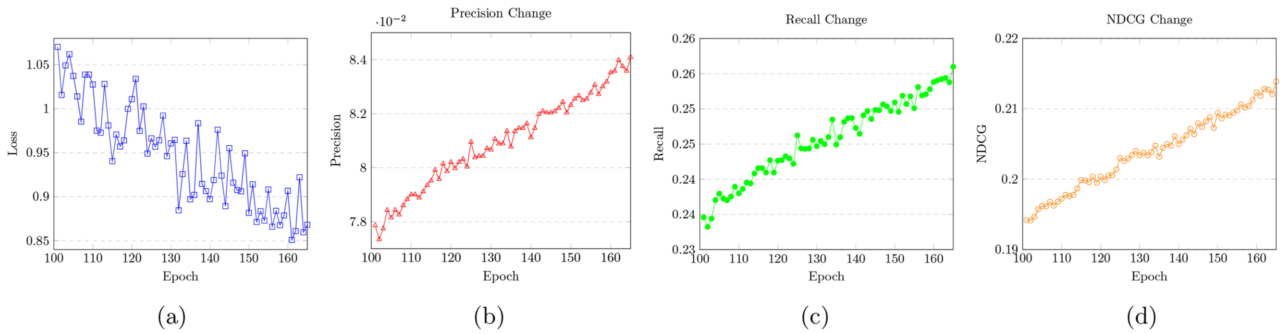


Fig. 4. Training loss and evaluation metrics evolution over epochs on Last FM dataset: (a) loss, (b) precision, (c) recall, and (d) NDCG.

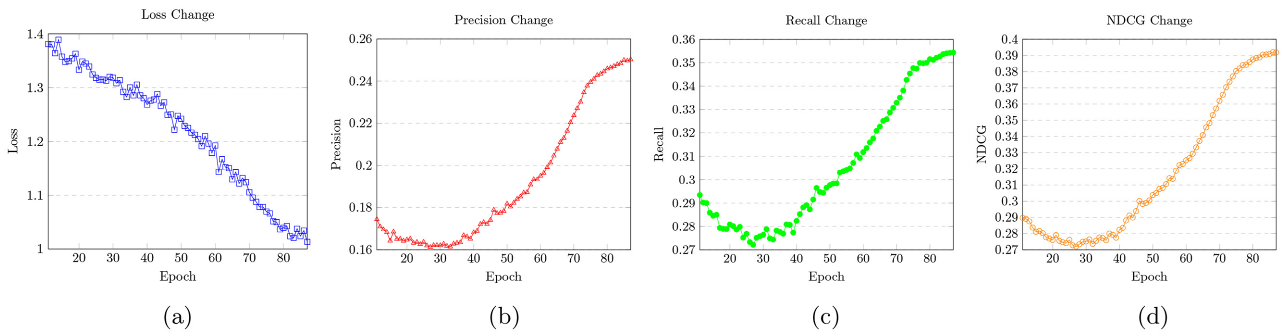


Fig. 5. Training loss and evaluation metrics evolution over epochs on MovieLens dataset: (a) loss, (b) precision, (c) recall, and (d) NDCG.

Table 2. Comparison of performance metrics for KHG-Aclair and baseline models (KGCL, KGRec) across Amazon-Book, Last FM, and MovieLens datasets

	Amazon-Book			**Last FM**			**MovieLens**		
	Recall	**Precision**	**NDCG**	**Recall**	**Precision**	**NDCG**	**Recall**	**Precision**	**NDCG**
KGCL	0.4289	0.0269	0.2396	0.2472	0.0786	0.1998	0.2486	0.1415	0.2388
KGRec	0.3834	0.0237	0.1962	0.2241	0.0717	0.1709	**0.3569**	**0.2557**	0.3863
KHG-Aclair	**0.4468**	**0.0282**	**0.2673**	**0.2560**	**0.0841**	**0.2139**	0.3544	0.2503	**0.3920**

The best performance for each metric is highlighted in bold.

1) Amazon-Book dataset

As can be seen in Table 2, In the Amazon-Book dataset, the KHG-Aclair model outperformed the other two models in all evaluation metrics. For Recall, KHG-Aclair scored approximately 3.5% higher than KGCL and 15.8% higher than KGRec. For Precision, KHG-Aclair surpassed KGCL by about 4.3% and KGRec by about 18.2%. For NDCG, KHG-Aclair was about 7.1% higher than KGCL and 30.9% higher than KGRec.

This superior performance of the KHG-Aclair model, which utilizes a knowledge hypergraph, can be attributed to the unique characteristics of the Amazon-Book dataset. The knowledge hypergraph effectively captures complex relationships between different book items such as authors, genres, and other information. These multifaceted connections allow for a more nuanced understanding of user preferences and item similarities. As a result, the model excels in retrieving relevant items, thus ensuring that there is a high proportion of relevant items among those retrieved, as well as in ranking these items appropriately, which leads to improved recommendation quality.

2) Last FM dataset

As can be seen in Table 2, in the Last FM dataset, the KHG-Aclair model also demonstrated superior performance across all metrics. For Recall, KHG-Aclair was about 2.5% higher than KGCL and 13.1% higher than KGRec. For Precision, KHG-Aclair was about 6.1% higher than KGCL and 16.3% higher than KGRec. For NDCG, KHG-

Aclair surpassed KGCL by about 6.2% and KGRec by 24.2%.

The enhanced performance of the KHG-Aclair model here can be attributed to the fact that the model is particularly well-suited to the Last FM dataset due to its intricate structure. The knowledge hypergraph has the ability to model the relationship between music items' diverse musical genres and artist connections. Understanding these complex, multi-dimensional relationships allow it to capture deeper insights into user preferences and item similarities. This shows that the KHG-Aclair model provides more accurate music recommendations and ranks them more effectively in the Last FM dataset.

3) MovieLens dataset

As can be seen in Table 2, in the MovieLens dataset, while KHG-Aclair achieved the highest performance in NDCG, KGRec slightly outperformed it in both Recall and Precision. For Recall, KHG-Aclair was about 0.9% lower than KGRec but 42.3% higher than KGCL. For Precision, KHG-Aclair was about 2.6% lower than KGRec but 76.0% higher than KGCL. For NDCG, KHG-Aclair was about 1.4% higher than KGRec and 64.1% higher than KGCL.

Despite its slightly lower Recall and Precision scores compared to KGRec, the higher NDCG score of KHG-Aclair indicates that it provides the best ranking quality of recommendations in the MovieLens dataset. This indicates that the KHG-Aclair model ultimately provides the best ranking quality of recommendations in the MovieLens dataset, thus demonstrating its superior overall recommendation quality compared to KGRec.

4) Summary

Overall, the KHG-Aclair model demonstrated the best performance across all evaluation metrics in the Amazon-Book and Last FM datasets, while it achieved the highest performance in NDCG in the MovieLens dataset. These results indicate that the KHG-Aclair model consistently delivers high performance across various domains, particularly in terms of recommendation quality and ranking. This underscores the KHG-Aclair model's effectiveness in enhancing user experience and providing more accurate and relevant recommendations. The superior performance of our model can be attributed to its ability to capture complex real-world relationships through the use of a hypergraph. By leveraging contrastive learning, the model benefits from improved representation learning, which enhances its recommendation capabilities. Taken together, these results confirm the effectiveness of our approach in developing a robust recommendation system.

C. Training Behavior Analysis of KHG-Aclair Model

It is crucial to visualize the optimization of loss value and evaluation metrics during the training process to further improve the model. In this section, we analyze the training behavior of the KHG-Aclair model to understand its performance and identify potential improvements.

1) Amazon-Book

The model's loss consistently decreased during the initial epochs and then stabilized after epoch 90. This trend indicates effective parameter tuning, reduced errors, and improved generalization over time. Precision slightly increased from approximately 0.026 initially to about 0.028 later on, indicating the model's enhanced ability to predict the positive class more accurately and reduce false positives. Recall rose from about 0.425 initially to around 0.45 later, suggesting improved accuracy in predicting a higher proportion of the actual positive class. Lastly, NDCG improved from around 0.23 initially to about 0.27 later, demonstrating better performance in ranking predictions as well as in accurately reflecting user preferences and relevance. These gradual improvements in the precision, recall, and NDCG metrics indicate that the model effectively learns and adapts over time. The continued fine-tuning of model parameters after stabilization in loss post-epoch 90 presents the possibility of optimizing performance without overfitting.

2) Last FM

The KHG-Aclair model demonstrates consistent improvements across various evaluation metrics on the Last FM dataset, showing that it effectively learns from data to enhance its recommendation performance. Starting from initially high loss values, it gradually decreased, thus indicating effective learning and parameter adjustment to minimize prediction errors. Precision started around 0.078 and increased progressively to about 0.084 by epoch 165. This improvement signifies the model's enhanced accuracy in predicting relevant positive cases. The recall metric began around 0.235 and rose to approximately 0.256 by epoch 165, indicating the model's improvement in identifying actual positive instances. NDCG, which is a crucial metric for evaluating ranking prediction quality, started around 0.194 and increased to about 0.214 by epoch 165. This continual enhancement reflects the model's ability to improve recommendation rankings, which is essential for an effective recommendation system. In conclusion, the KHG-Aclair model demonstrates promising improvements in loss, precision, recall, and NDCG metrics on the Last FM dataset, altogether showcasing its effective learning and adaptation capabilities.

3) MovieLens

The KHG-Aclair model demonstrated continuous improvement across various metrics during the training process on the MovieLens dataset, indicating effective learning and enhancement in recommendation performance. The model initially had high loss values, but its loss was gradually reduced as the number of epochs increased, thus indicating effective data assimilation. Moreover, the precision, which was initially low, increased over time, indicating that the model is providing more relevant recommendations to users. The recall also improved, indicating that the model enhanced its ability to identify and recommend more items compared to its early stages, implying a greater inclusion of movies that users are likely to prefer. The NDCG metric also showed improvement compared to its initial values, indicating the model's efforts to enhance the ranking of recommendation results and increase user satisfaction. Overall, the KHG-Aclair model consistently improved across metrics such as loss, precision, recall, and NDCG on the MovieLens dataset. These results demonstrate the model's effectiveness in learning from diverse movie data and optimizing recommendation performance.

4) Summary

The training results of the KHG-Aclair model across three datasets indicate consistent improvement in all metrics as the number of epochs increases. This suggests that the model learns progressively and enhances its prediction accuracy and ranking quality over time. These results support the superior learning ability of the KHG-Aclair model and its applicability across diverse datasets.

VII. FUTURE DIRECTIONS

For future research, while our current implementation generates the knowledge hypergraph based on 1-hop connections for computational efficiency, we plan to extend and refine the knowledge hypergraph by including larger and more complex structures. We will also ensure that these enhanced knowledge hypergraphs maintain computational efficiency. We will also integrate contextual features such as user demographics, temporal dynamics, or product attributes to enrich recommendation accuracy and relevance. This expansion will further enhance the recommendation system's learning and performance, thus paving the way for more sophisticated and accurate recommendations.

VIII. CONCLUSION

In conclusion, our proposed KHG-Aclair system addresses challenges in recommendation systems by leveraging knowledge hypergraphs to capture intricate relationships that are often overlooked when using traditional methods. This approach enhances recommendation accuracy through an attention mechanism and contrastive learning. By transforming the Freebase into a knowledge hypergraph and making it publicly available, we demonstrate significant advancements in recommendation performance that are expected to serve as valuable resources for future research and development. The KHG-Aclair model consistently excels across diverse datasets, thus underscoring its versatility and potential for various recommendation scenarios. This adaptability enhances user experience by delivering precise and relevant recommendations across domains.

CONFLICT OF INTEREST

The authors have declared that no competing interests exist.

ACKNOWLEDGMENTS

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the government of Korea (MSIT) (No.2021-0-01352, Development of technology for validating the autonomous driving services in perspective of laws and regulations).

REFERENCES

1. W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, “Graph neural networks for social recommendation,” in *Proceedings of the World Wide Web Conference*, San Francisco, CA, USA, 2019, pp. 417-426. https://doi.org/10.1145/3308558.3313488
2. C. Huang, “Recent advances in heterogeneous relation learning for recommendation,” 2021 [Online]. Available: https://arxiv.org/abs/2110.03455.
3. R. Togashi, M. Otani, and S. I. Satoh, “Alleviating cold-start problems in recommendation through pseudo-labelling over knowledge graph,” in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, Virtual Event, Israel, 2021, pp. 931-939. https://doi.org/10.1145/3437963.3441773
4. C. Huang, H. Xu, Y. Xu, P. Dai, L. Xia, M. Lu, et al., “Knowledge-aware coupled graph neural network for social recommendation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 4115-4122, 2021. https://doi.org/10.1609/aaai.v35i5.16533
5. X. Wang, X. He, Y. Cao, M. Liu, and T. S. Chua, “KGAT: knowledge graph attention network for recommendation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 2019, pp. 950-958. https://doi.org/10.1145/3292500.3330989
6. Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo, and Y. Zhang, “Reinforcement knowledge graph reasoning for explainable recommendation,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and*

Development in Information Retrieval, Paris, France, 2019, pp. 285-294. https://doi.org/10.1145/3331184.3331203

7. J. Wen, J. Li, Y. Mao, S. Chen, and R. Zhang, "On the representation and embedding of knowledge bases beyond binary relations," 2016 [Online]. Available: https://arxiv.org/abs/1604.08642.
8. B. Fatemi, P. Taslakian, D. Vazquez, and D. Poole, "Knowledge hypergraphs: prediction beyond binary relations," 2019 [Online]. Available: https://arxiv.org/abs/1906.00137.
9. Y. Yang, C. Huang, L. Xia, and C. Li, "Knowledge graph contrastive learning for recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Madrid, Spain, 2022, pp. 1434-1443. https://doi.org/10.1145/3477495.3532009
10. H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proceedings of the World Wide Web Conference*, San Francisco, CA, USA, 2019, pp. 3307-3313. https://doi.org/10.1145/3308558.3313417
11. X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, and T. S. Chua, "Learning intents behind interactions with knowledge graph for recommendation," in *Proceedings of the World Wide Web Conference*, Ljubljana, Slovenia, 2021, pp. 878-887. https://doi.org/10.1145/3442381.3450133
12. Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *Proceedings of the ACM Web Conference*, Virtual Event, Lyon, France, 2022, pp. 2320-2329. https://doi.org/10.1145/3485447.3512104
13. J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "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, 2021, pp. 726-735. https://doi.org/10.1145/3404835.3462862
14. W. Wei, C. Huang, L. Xia, and C. Zhang, "Multi-modal self-supervised learning for recommendation," in *Proceedings of the ACM Web Conference*, Austin, TX, USA, 2023, pp. 790-800. https://doi.org/10.1145/3543507.3583206
15. J. Yu, H. Yin, M. Gao, X. Xia, X. Zhang, and N. Q. Viet Hung, "Socially-aware self-supervised tri-training for recommendation," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Virtual Event, Singapore, 2021, pp. 2084-2092. https://doi.org/10.1145/3447548.3467340
16. Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 3558-3565, 2019. https://doi.org/10.1609/aaai.v33i01.33013558
17. W. X. Zhao, G. He, K. Yang, H. Dou, J. Huang, S. Ouyang, and J. R. Wen, "KB4Rec: a data set for linking knowledge bases with recommender systems," *Data Intelligence*, vol. 1, no. 2, pp. 121-136, 2019. https://doi.org/10.1162/dint_a_00008
18. K. Ding, J. Wang, J. Li, D. Li, and H. Liu, "Be more with less: hypergraph attention networks for inductive text classification," 2020 [Online]. Available: https://arxiv.org/abs/2011.00387.
19. M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu, "Dimensionality reduction for k-means clustering and low rank approximation," in *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, Portland, OR, USA, 2015, pp. 163-172. https://doi.org/10.1145/2746539.2746569
20. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012 [Online]. Available: https://arxiv.org/abs/1205.2618.
21. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *Proceedings of Machine Learning Research*, vol. 119, pp. 1597-1607, 2020.
22. T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: a next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 2019, pp. 2623-2631. https://doi.org/10.1145/3292500.3330701
23. Y. Yang, C. Huang, L. Xia, and C. Huang, "Knowledge graph self-supervised rationalization for recommendation," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Long Beach, CA, USA, 2023, pp. 3046-3056. https://doi.org/10.1145/3580305.3599400

Hyejin Park (https://orcid.org/0009-0005-4847-2116)

Hyejin Park received her master's degree in Computer Science from Yonsei University, Seoul, Korea, in 2022. Since then, she has been working as a Researcher at the Korea Electronics Technology Institute in Seoul, Korea. Her research interests focus on (Hyper)Graph Learning and utilizing graphs for information retrieval and recommendation systems.

Taeyoon Lee (https://orcid.org/0000-0002-9757-8882)

Taeyoon Lee received her master's degree in Computer Science from Yonsei University, Seoul, Korea, in February 2023. She has been working as a Researcher at the Korea Electronics Technology Institute in Seoul, Korea. Her research interests include recommendation systems and data analysis.

Kyungwon Kim (https://orcid.org/0000-0001-9537-9225)

Kyungwon Kim received Ph.D. degree in Computer, Information and Communications Engineering from Konkuk University, Seoul, Korea, in March 2018. He has been a Principal Researcher at Korea Electronics Technology Institute in Seoul, Korea, since 2004. His current research interests include unstructured data analysis and data inference modeling.