

Article

Not peer-reviewed version

A Global Structural Hypergraph Convolution Model for Bundle Recommendation

[Xingtong Liu](#) * and [Man Yuan](#) *

Posted Date: 8 August 2023

doi: 10.20944/preprints202308.0525.v1

Keywords: recommender system; bundle recommendation; hypergraph neural network; hypergraph



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

A Global Structural Hypergraph Convolution Model for Bundle Recommendation

Xingtong Liu *  and ManYuan *

School of Computer & Information Technology, Northeast Petroleum University, Fazhan Road, Daqing 163318, China

* Correspondence: lxtidyx2022@163.com (X.L.); yuanman@nepu.edu.cn (M.Y.)

Abstract: Bundle recommendation is provided to users by combining related items into a bundle, enhancing the user's shopping experience and increase the merchant's sales revenue. Despite the success of existing solutions based on Graph Neural Networks(GNN), there are still some significant challenges: (1)It is demanding to explicitly model multiple complex associations using standard graph neural networks. (2)Numerous additional nodes and edges are introduced to approximate higher-order associations. (3)The user-bundle historical interaction data is highly sparse. In this work, we propose a Global Structural Hypergraph Convolution model for Bundle Recommendation (SHCBR) to address the above problems. Specifically, we jointly incorporate multiple complex interactions between users, items, and bundles(three entities) into a relational hypergraph without introducing additional nodes and edges. Higher-order associations are already involved in the hypergraph structure which alleviate the training burden of neural networks and the dilemma of scarce data. In addition, we design a special matrix propagation rule that captures non-pairwise complex relationships between entities. Using item nodes as links, hypergraph convolution propagation between user and bundle nodes generates the learned representations. Experimental results on two real-world datasets demonstrate that SHCBR outperforms the state-of-the-art baselines by 11.07%–25.66% on Recall and 16.81%–33.53% on NDCG. We publicly release the codes and datasets at <https://github.com/Lxtdzh/SHCBR-master/>.

Keywords: recommender system; bundle recommendation; hypergraph neural network; hypergraph

1. Introduction

With the rapid development of the e-commerce industry, the recommender system has become an essential tool to improve user experience and promote product sales [1,2]. Traditional recommender systems mainly focus on recommending individual items, which couldn't meet the growing personalized needs of users. To further enhance user satisfaction, bundle recommendation as a marketing strategy has been proposed. Based on user's purchasing behavior and item relevance, bundle recommendation combines relevant items into bundles, such as music playlists [3], game bundles [4,5], and drug packages [6]. Recommending bundles of related items can provide users with more favorable personalized combination choices, while also benefiting businesses by making profits [7].

The core of recommender systems is to predict how likely a user will interact with an item, such as making a purchase or clicking on it [8]. However, the decision-making process for a user to choose bundles is much more complex than selecting individual items. Specifically, users will simultaneously consider multiple items in the bundle and their combination discounts. Sometimes, even if users like all the items contained, they may not prefer the bundle because it's not a satisfactory well-matched combination. Only when users are most satisfied with the combination of items in the bundle could they choose the bundle for consumption. Therefore, user-bundle interactions are usually sparser than user-item interactions. At the same time, the user-item-bundle relationship is much more complex than pairwise relationship. For example, a user can interact with multiple items or bundles, and an item can

exist in multiple bundles. In this case, the affinity relations are no longer dyadic (pairwise), but rather integration of multiple binary relationships between three entities. Accordingly, the aforementioned issues make the bundle recommendation task highly challenging.

In recent years, Graph Neural Network(GNN) [9], a type of neural network designed to process and learn from graph-structured data, has become one of the research hotspots. The idea behind GNN [10,11] is to perform message passing between neighboring nodes in the graph to learn node representations based on their local neighborhood information. This allows GNN to capture complex and non-linear relationships between nodes in the graph. Currently, there are some challenges in using GNN to model bundle recommendation tasks. Specifically, due to the interactions between users and items/bundles, as well as the affiliation relationships between items and bundles, it is difficult to explicitly model the multiple complex associations among users, items, and bundles(three entities) using standard graph neural networks. Furthermore, traditional graphs are limited in the sense that each edge can only connect two nodes. As a result, a large number of additional nodes and edges need to be introduced to approximate higher-order associations between entities, which increases the training burden on neural networks. Finally, recommender systems generally suffer from the problem of sparse user-item interaction history data. In bundle recommendation scenarios, the data sparsity issue is even more prominent because bundles contain multiple items, resulting in naturally fewer user-bundle interaction history data compared to user-item interaction history data. The highly sparse historical interaction data makes learning with neural networks difficult and unstable. This also results in dilemma for bundle recommendation models to accurately model user preferences.

Hypergraph [12], as a more flexible graph structure compared to traditional graphs, provides a natural solution to address above limitations. As shown in Figure 1, in traditional graphs, an edge can only connect two nodes, while a hyperedge in hypergraphs can connect multiple vertices simultaneously. Therefore, hypergraphs can more flexibly represent and capture complex relationships in real applications. Consequently, some problems could be easier to solve with the more accurate representation provided by hypergraphs.

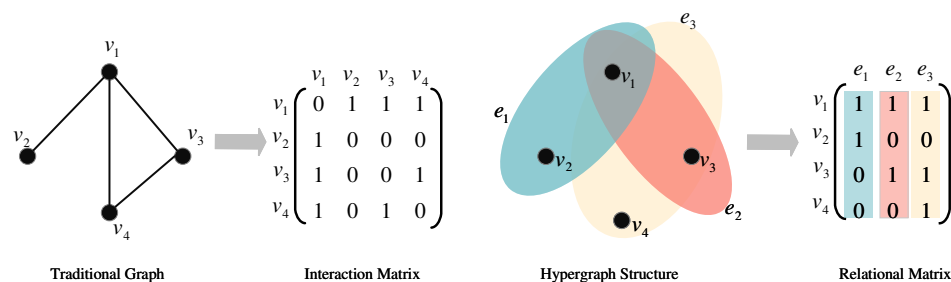


Figure 1. The difference between a traditional graph and a hypergraph. In a traditional graph, each edge, denoted by a line, only connects two vertices. In a hypergraph, each edge, denoted by a colored ellipse, can connect more than two vertices.

In this work, we propose a novel Global Structural Hypergraph Convolution model for Bundle Recommendation(SHCBR), which jointly incorporate multiple complex interactions between users, items, and bundles into a relational hypergraph. We directly connect user nodes, item nodes and bundle nodes(three types of nodes) with a hyperedge, which explicitly models complex associations among the three entities without introducing additional nodes and edges. Further, we design a special matrix propagation rule that uses items as links to aggregate and update the embeddings of users and bundles on the hypergraph. Additionally, we design a personalized weight operation to improve the accuracy of the final recommendation results. Meanwhile, inspired by LightGCN [13], we simplify the original hypergraph convolution by removing feature transformations and non-linear activation functions, making it more suitable for bundle recommendation scenarios. Hypergraph

convolution is able to learn the hidden layer representation considering the high-order information, thereby improving the quality of user and bundle representation learning. SHCBR further integrates the learned representations to generate recommendations by ranking the prediction scores.

To summarize, the main contributions of this work are as follows:

- We propose a novel model named SHCBR, which introduces hypergraph structure to explicitly model complex relationships between users, items and bundles in bundle recommendation tasks. We directly use three types of nodes with a hyperedge without introducing additional nodes and edges. By constructing a relational hypergraph containing three types of nodes, we can explore existing information from a global perspective, effectively alleviating the dilemma of data sparsity;
- we design a special matrix propagation and personalized weight operation in proposed structural hypergraph convolutional neural network (SHCNN). Using items as links, we leverage efficient hypergraph convolution to learn the hidden layer representation considering the high-order information. Since higher-order associations are already involved in the hypergraph structure, a single layer of SHCNN is sufficient to capture node representations, which further improves model efficiency;
- Extensive experiments on two real-world datasets show that our proposed approach outperforms existing state-of-the-art baselines by 11.07%-25.66% on Recall and 16.81%-33.53% on NDCG.

2. Related Work

In this section, we briefly review the related works on bundle recommendation and hypergraph learning.

2.1. Bundle Recommendation

Despite the extensive research on recommender systems, few efforts have been devoted to addressing the specific challenges of bundle recommendation tasks. In earlier years, List Recommendation Model (LIRE) [14] explored a latent factor-based BPR [15] model to recommend item lists, which simultaneously took into consideration users' previous interactions with both item lists and individual items. Embedding Factor Model (EFM) [3] combined Word2VEC [16] and the latent factor model to enhance recommendation performance, with jointly modeling user-item and user-list interactions. Deep Attentive Multi-Task (DAM) [17] model utilized a factorized attention network to aggregate the item embeddings in a bundle to obtain the bundle's representation. Simultaneously, DAM jointly modeled user-bundle interactions and user-item interactions in a multi-task manner to alleviate the scarcity of user-bundle interactions.

In recent years, graph neural networks have gained widespread attention in the field of recommender systems due to their superior representation and modeling capabilities for graph-structured data. Furthermore, GNN-based models achieved superior performance in bundle recommendation tasks. Bundle Graph Convolutional Network (BGCN) [18] unified three interactions among users, items, and bundles into a heterogeneous graph and used graph convolutional networks (GCN) [19] to learn the representation of users and bundles capturing item level semantics. BundleNet [5] formalized a recommendation task as a link prediction problem on a user-item-bundle tripartite graph constructed from the historical interactions, and tackled it with a neural network model that can learn directly on the graph-structure data. Multi-view Intent Disentangle Graph Networks (MIDGN) [20] model disentangled user-item and bundle-item graph coupling with the user's intents from global and local views respectively, with the help of GNN equipped with the neighbor routing mechanism. Meanwhile, MIDGN also compared the intents learned from different views to better represent the user's preference as well as the items' associations at a finer grain of the user's intent. Interactive Multi-Relation Bundle Recommendation with Graph Neural Network (IMBR) [21] used the graph neural network to extract different representations of multi-relation from different

views and designed a Bundle Frequent Term Constraint(BFTC) algorithm to obtain more accurate bundle representations.

Most of the state-of-the-art works modeled the interactions between users, items, and bundles on traditional graph data structures. However, due to the complexity and sparsity of user-bundle interactions, there are some significant limitations to model the interactions between entities in bundle recommendation tasks. This brings back the concept of hypergraph [12], a special graph model which leverages hyperedges to connect multiple vertices simultaneously.

2.2. Hypergraph learning

Hypergraph [12,22] is a special graph structure that extends traditional graphs. It allows a hyperedge to connect multiple nodes, forming a many-to-many relationship. Therefore, hypergraphs can more accurately represent complex relationships that traditional graphs cannot directly capture in real-world scenarios. For example, in the knowledge graph [23], hypergraphs can represent relationships between an entity and multiple other entities. Such higher-order associations are useful for modeling complex semantic associations. In contrast, traditional graphs would require the introduction of a large number of nodes and edges to approximate such higher-order associations.

Hypergraph learning [24] is a deep learning method based on hypergraph structures. Due to its flexibility and ability to model complex data associations, hypergraph learning has attracted increasing attention in recent years. Hypergraph learning was initially proposed as a label propagation method [25] for semi-supervised learning. This method aimed to minimize the label differences among vertices sharing the same hyperedge. Huang et al. [26] applied hypergraph learning to video object segmentation and discussed the construction methods of hypergraphs. Weights have a great influence on the modeling of data correlation, and then learning the weights of hyperedges has become a new research topic. Gao et al. [27] introduced a multi-hypergraph structure to assign weights to different sub-hypergraphs, with the intention of assigning larger weights to hyperedges or sub-hypergraphs with higher importance. Subsequently, the L2 regularization of weights to learn optimal hyperedge weights was proposed by [28]. Like graph neural networks, the hypergraph neural networks(HGNN) [29] has been proposed as the first deep learning method on hypergraph structure. Hypergraph neural networks employed the hypergraph Laplacian operator to represent the hypergraph from a spectral perspective. Compared to the existing methods based on graph neural networks, hypergraph neural networks can naturally model high-order relationship among data, which is effectively exploited and encoded to facilitate feature extraction. Subsequently, the authors of [30] introduced two end-to-end trainable operators, hypergraph convolution and hypergraph attention, which can handle non-pairwise relationship modeled in a high-order hypergraph. With the two operators, a graph neural network can be readily extended to a more flexible model and applied to diverse applications where non-pairwise relationships are observed.

Although research on hypergraph deep learning is still in its early stages, hypergraph neural networks have been widely applied in various fields [23,31–34] due to their exceptional representation capabilities. For example, in computer vision, Wang et al. [31] applied hypergraphs to visual classification tasks, using hypergraphs to describe the relations among visual features. In knowledge graphs [23], the hypergraph neural networks were leveraged to model the correlation between entities and attributes at a higher level. In the field of recommender systems, hypergraph-based recommendation models can capture complex interactions between entities and provide more accurate and diverse recommendation results. For instance, Hg-PDC [33] proposed a collaborative filtering recommendation algorithm based on dynamics clustering and similarity measurement in hypergraphs, thereby improving the recommendation performance. The authors of [34] proposed a global context-supported hypergraph enhanced graph neural network(GC-HGNN) for session-based recommendation, which utilized hypergraph convolutional neural network to obtain the global context information.

3. Preliminary

In this section, we briefly formulate the problem and introduce the bundle hypergraph definition.

3.1. Problem Formulation

Given a set of users $U = \{u_1, u_2, \dots, u_N\}$, a set of items $I = \{i_1, i_2, \dots, i_M\}$, and a set of bundles $B = \{b_1, b_2, \dots, b_K\}$, where N , M , and K are the number of users, items, and bundles, respectively. For each bundle $b_s \in B$, it consists of a set of items, $b_s = \{i_{s_1}, i_{s_2}, \dots, i_{s_{|b_s|}}\}$, where $|b_s|$ denotes the bundle size (larger than 1). We define user-item matrix, user-bundle matrix and bundle-item matrix as $X_{N \times M} = \{x_{ui} | u \in U, i \in I\}$, $Y_{N \times K} = \{y_{ub} | u \in U, b \in B\}$, and $Z_{K \times M} = \{z_{bi} | b \in B, i \in I\}$ with a binary value at each entry, respectively. An observed interaction $x_{ui} = 1$ means user u once interacted item i , and an observed interaction $y_{ub} = 1$ means user u once interacted bundle b . Similarly, an entry $z_{bi} = 1$ means bundle b contains item i . Based on the above definition, the problem of bundle recommendation is then formulated as follows:

Input : user-bundle interaction data $X_{N \times M}$, user-item interaction data $Y_{N \times K}$, and bundle-item affiliation data $Z_{K \times M}$.

Output : A recommendation model that estimates the probability that user u will interact with bundle b .

3.2. Bundle Hypergraph Definition

Hypergraphs provide a natural way to capture complex higher-order connections. However, traditional hypergraphs in existing methods struggle to represent multiple relationships in bundle recommendation tasks suitably. In response, we propose a data structure called the bundle hypergraph, which can more effectively capture the multiple associations among users, items, and bundles in bundle recommendations.

A bundled hypergraph can be defined as a mathematical structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of nodes, and \mathcal{E} is the set of hyperedges. Each hyperedge can contain multiple vertices and is represented as a subset of the vertex set. This definition allows for a more flexible representation of relationships between vertices compared to traditional graphs. Let the incidence matrix $H_{ve} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ denote the association relationship in hypergraph, where the entry $h(v, e)$ can be defined as:

$$h(v, e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e \end{cases} \quad (1)$$

According to the definition of hypergraph, we can calculate the degree of vertex and hyperedge. Then, for a vertex $v \in \mathcal{V}$, the degree is defined as $d_v = \sum_{e \in \mathcal{E}} h(v, e)$. For a hyperedge $e \in \mathcal{E}$, the degree is defined as $d_e = \sum_{v \in \mathcal{V}} h(v, e)$. Further, we denote two diagonal matrices $D_v \in \mathbb{N}^{|\mathcal{V}| \times |\mathcal{V}|}$ and $D_e \in \mathbb{N}^{|\mathcal{E}| \times |\mathcal{E}|}$ to represent the degree of nodes and hyperedges, respectively.

4. Method

In this section, we first present the construction of the relational hypergraph in Section 4.1. Then, in Section 4.2, we elaborate on the structural hypergraph convolutional neural network. In Section 4.3, we introduce the matrix propagation rules. Finally, we describe the model prediction and training in Section 4.4.

4.1. Relational Hypergraph Construction

As shown in Figure 2, we construct a relational hypergraph matrix with items as links. To obtain more meaningful bundle representations, we first construct three types of relation graphs around users and bundles. These include the user-item interaction graph, the user-bundle interaction graph, and the bundle-item affiliation graph. Based on these three relation graphs, we can obtain three interaction matrices mentioned in the Section 3.1, namely X , Y , and Z . For each node on the hypergraph, we apply

one-hot encoding to encode the input and compress them to dense real-value vectors. We use E_u , E_i , E_b to denote the initial user embedding matrix, item embedding matrix and bundle embedding matrix respectively, which have the same embedding size d . The feature vectors for user u , item i , and bundle b are defined as follows:

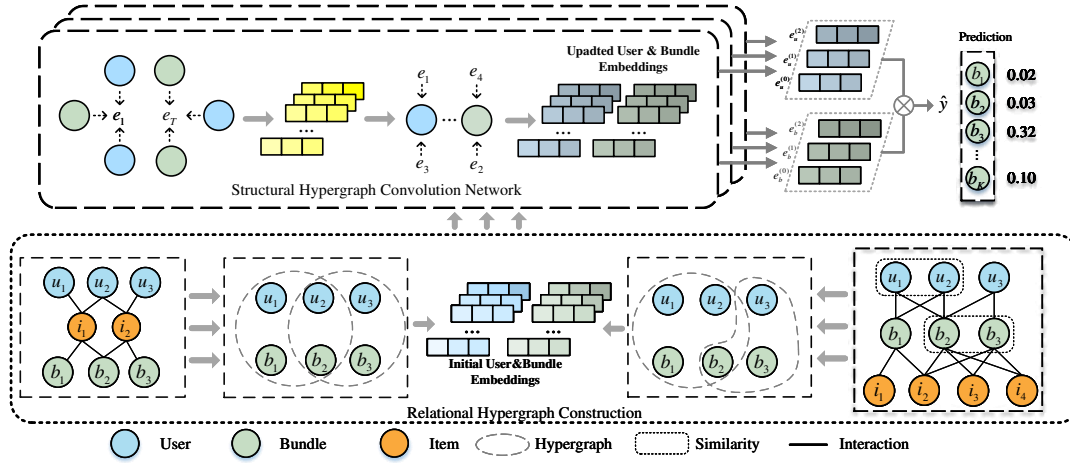


Figure 2. The overall framework of SHCBR consists of two parts: (1) Relational Hypergraph Construction, which constructs a relational hypergraph matrix with items as links and obtains user/bundle initial embeddings; (2) The Structural Hypergraph Convolutional Neural Network is our proposed for capturing high-order relationships of users and bundles in hypergraphs.

$$e_u = E_u^T v_u, \quad e_i = E_i^T v_i, \quad e_b = E_b^T v_b \quad (2)$$

where $v_u \in \mathbb{R}^N$, $v_i \in \mathbb{R}^M$ and $v_b \in \mathbb{R}^K$ denote the one-hot feature vector for user u , item i , and bundle b , respectively.

According to the definition in Section 3.1, in the application scenario of bundle recommendation, we naturally get user-item interaction matrix X , user-bundle interaction matrix Y , and bundle-item affiliation matrix Z . Given the complexity of relationships among the three types of nodes, the relevant items can serve as links between users and bundles. we construct the item association matrix as follows:

$$H_I = \begin{bmatrix} X & Z \end{bmatrix}^T \quad (3)$$

where $X \in \mathbb{R}^{|U| \times |I|}$, $Z \in \mathbb{R}^{|B| \times |I|}$. Similarly, using items as connectors and leveraging second-order interaction data, we can infer the similarity between users. Likewise, we infer the similarity between bundles. Highly overlapping users and bundles aid in predicting user interests in bundles. Therefore, we further define the user-bundle adjacency matrix as follows:

$$H_{UB} = \begin{bmatrix} S_U & Y \\ Y^T & S_B \end{bmatrix} \quad (4)$$

$$S_U = X \cdot X^T, \quad S_B = Z \cdot Z^T \quad (5)$$

where $S_U \in \mathbb{R}^{|U| \times |U|}$ and $S_B \in \mathbb{R}^{|B| \times |B|}$ denote the user similarity-overlap matrix and the bundle similarity-overlap matrix constructed based on second-order interactions, respectively.

Based on the flexibility of hypergraphs in representing and capturing complex relationships in practical problems, the relational hypergraph matrix we construct can be defined as follows:

$$H_{SC} = \begin{bmatrix} H_{UB} & H_I \end{bmatrix} \quad (6)$$

where $H_{SC} \in \mathbb{R}^{(|U|+|B|) \times (|U|+|B|+|I|)}$. The three types of nodes are jointly incorporated in a relational hypergraph H_{SC} from a global perspective, naturally capturing higher-order connections within it.

4.2. Structural Hypergraph Convolutional Neural Networks

We propose a structural hypergraph convolutional neural network (SHCNN) as shown in Figure 2, which captures the higher-order associations between three entities on the hypergraph structure. Hypergraphs [12] are an extension of graphs where hyperedges can connect multiple nodes. Compared with traditional graph structures, hypergraphs have more complex connectivity, making them more suitable for bundle recommendation tasks. Like graph convolutional networks, hypergraph convolution [30] can also be regarded as a message passing mechanism. The message passing paths of hypergraphs are more complex compared to traditional graph neural networks. How to define the convolution operation and efficiently propagate information between adjacent nodes is very crucial in the bundle recommendation system. Increasing the number of convolutional layers in graph neural networks can naturally capture high-order association information. However, adding more convolutional layers also leads to a dramatic increase in computational cost. Since higher-order associations are already involved in the hypergraph structure, a single layer of SHCNN is sufficient to learn node representations. We further simplify the formula for hypergraph convolution operation as follows:

$$\hat{H} = D_v^{-\frac{1}{2}} H D_e^{-\frac{1}{2}} H^T D_v^{-\frac{1}{2}} \quad (7)$$

$$E^{(l+1)} = \hat{H} E^{(l)} \quad (8)$$

where \hat{H} refers to the Laplacian spectral normalization of relational hypergraph matrix. $E^{(l+1)}$ is the output of layer l .

Furthermore, to learn the representations of users and bundles more accurately, we design a personalized weight operation. Inspired by the attention mechanism [35] in neural networks, we make improvements to the similarity-overlap matrix in Equation 5. Taking S_U as an example, we can compute the similarity between user u_i and user u_j as follows:

$$S_{ij} = \frac{\exp(\sigma(\text{sim}(u_i, u_j)))}{\sum_{k \in N} \exp(\sigma(\text{sim}(u_i, u_k)))} \quad (9)$$

where $S_{ij} \in S_U$. $\sigma(\cdot)$ is the nonlinear activation function like LeakyReLU [36] and eLU [37], enhancing the learning ability of neural networks. Likewise, we compute the similarity between bundles. Through the above computations, we can get the user similarity-overlap matrix S_U and the bundle similarity-overlap matrix S_B .

4.3. Matrix Propagation Rules

Combining the application scenario of bundle recommendation and the SHCBR model, we propose a specialized matrix propagation rule to achieve the embedding propagation of the entire relational hypergraph. In order to simulate the complex interaction logic among users, items, and bundles, we define the relational hyperedge adjacency matrix as follows:

$$A = H_{SC} D_e^{-\frac{1}{2}} H_{SC}^T \quad (10)$$

where H_{SC} is the relational hypergraph matrix we constructed in Equation 6. The Laplacian matrix L based on the relational hypergraph matrix H_{SC} can be defined as:

$$D_v = \begin{bmatrix} D_U & 0 \\ 0 & D_B \end{bmatrix} \quad (11)$$

$$L = D_v^{-1/2} A D_v^{-1/2} + \begin{bmatrix} I_U & 0 \\ 0 & I_B \end{bmatrix} = \begin{bmatrix} I_U & D_U^{-1/2} A_U D_B^{-1/2} \\ D_B^{-1/2} A_B D_U^{-1/2} & I_B \end{bmatrix} \quad (12)$$

where A_U and A_B are the relational hyperedge adjacency matrix of users and bundles respectively. A_U and A_B together constitute the relational hyperedge adjacency matrix A . D_U and D_B are the node degree matrices of users and bundles, respectively. $I_U \in \mathbb{R}^{|U| \times |U|}$ and $I_B \in \mathbb{R}^{|B| \times |B|}$ are the identity matrix. Inspired by LightGCN, the purpose of adding the identity matrix is to connect its own nodes.

4.4. Model Prediction and Training

In the field of hypergraph convolution, traditional methods like hypergraph neural network [29] use the embeddings from the last layer as the final representation. However, as the number of layers increases, the embeddings tend to become over-smoothing, making it unreasonable to use only the last layer embeddings as the final representation. In existing bundle recommendation models, BGCN [18] concatenates all layers' embeddings to combine the information received from neighbors of different depths for prediction, which would exacerbate the training difficulty. Inspired by LightGCN [13], we choose to incorporate embeddings from different layers as the final representation. This approach captures the information from different layers to enrich the semantics of the final representation. Next, we combine the embeddings of each layer to obtain the final representation of users and bundles:

$$e_u = \sum_{k=0}^K \alpha_k e_u^{(k)}, e_b = \sum_{k=0}^K \alpha_k e_b^{(k)} \quad (13)$$

where $\alpha_k \geq 0$. $\alpha_k \geq 0$ denotes the importance of the k -th layer embedding in constituting the final embedding. We setting α_k as $1/(K+1)$.

Finally, we define the inner production of user and bundle final representations as the model prediction:

$$\hat{y}_{ub} = e_u^T e_b \quad (14)$$

The final results are used as the ranking scores for recommendation generation. Observations indicate that if a user interacts with a bundle (e.g., purchase, click), we can assume that the user is interested in the bundle or most of its items. Conversely, if a user does not interact with a bundle, it can be assumed that the user is not aware of that bundle. Therefore, we consider the bundles with interactions as positive samples and randomly select an unobserved bundle as a negative sample. Finally, we define a pairwise learning framework and use the Bayesian Personalized Ranking (BPR) loss for bundle prediction:

$$L_{bundle} = (1 - \alpha) \sum_{(j,e) \in R^+} \ln \sigma(\hat{y}_{je} - y_{je}) + \alpha \sum_{(j,f) \in R^-} \ln \sigma(\hat{y}_{jf} - y_{jf}) \quad (15)$$

where $R = \{(j, e, f) \mid (j, e) \in R^+, (j, f) \in R^-\}$. R^+ denotes bundles that has interactions with user u_j , while R^- denotes bundles that has no interactions with user u_j . we regard (j, e) as a positive sample, and (j, f) as a negative sample. $\sigma(\cdot)$ denotes the sigmoid function. We use L_2 regularization in the loss function to prevent the model from overfitting.

Then, we use the AdamW optimizer [38] to minimize the loss function. The AdamW optimizer can update the weights of neural networks based on the training data iteratively.

5. Experiment

In this section, we conduct experiments on two real-world datasets to evaluate our proposed SHCBR, with the aim of answering following research questions:

- RQ1: How does our proposed SHCBR perform compared to previous models?
- RQ2: How do different components affect the performance of the SHCBR?

- RQ3: How do different parameter settings influence the results of the SHCBR?

5.1. Experiment Settings

5.1.1. Datasets & Metrics

We evaluate our proposed SHCBR and all baseline models on two real-world datasets. We randomly split the train/validation/test sets into a ratio of 70%/10%/20% with statistics shown in Table 1.

- **NetEase** This is a dataset constructed using data provided by a Chinese music platform, Netease Cloud Music (<http://music.163.com>). As a social music software, it allows users to freely choose their favorite songs and add them to their favorites. Users can also choose to listen to playlists bundled with different songs.
- **Youshu** This is a dataset constructed by the famous Chinese book review website Youshu (<https://www.yousuu.com/>). Similar to the NetEase dataset, it allows users to build lists of books they want, each of which is a bundle.

Table 1. Dataset Statistics.

Dataset	NetEase	Youshu
User	18,528	8,039
Item	22,864	4,771
Bundle	123,628	32,770
User-Bundle	302,303	51,337
User-Item	1,128,065	138,515
Bundle-Item	1,778,065	176,667
Avg. Bundle Interactions	16.32	6.39
Avg. Item Interactions	60.88	17.23
Avg. Bundle Size	77.80	37.03
User-Item Density	0.05%	0.05%
User-Bundle Density	0.07%	0.13%

To evaluate the recommendation performance of the models, we employ two widely adopted ranking metrics: Recall@K and normalized discounted cumulative gain(NDCG@K). Recall@K measures the ratio of test bundles that have been contained by the top-K ranking list. NDCG@K complements Recall by assigning higher scores to the hits that appear at higher positions in the list. It worth mentioning that we consider the following three different sizes of K values: {20, 40, 80}. Recall and NDCG are calculated as follows:

$$Recall@K = \frac{|D \cap B|}{D} \quad (16)$$

where K indicates that we perform top- K rank. We generate a bundle list B of length K for each user. D is a bundle set with which the user has already interacted. b_k indicates that the bundle at position k in the generated bundle list B , and d represents the bundle in bundle set D that the user has actually interacted with.

$$DCG@K = \sum_{k=1}^K \frac{2^{hit_k} - 1}{\log_2(k+1)} \quad (17)$$

where $hit_k = 1$ when $b_k \in D$, otherwise $hit_k = 0$.

$$NDCG@K = \frac{DCG@K}{IDCG@K} \quad (18)$$

where Ideal Discounted Cumulative Gain at K(IDCG@K) measures the relevance of the top K ranked items by comparing the discounted cumulative gain of the actual ranking to the ideal ranking.

5.1.2. Baselines

To demonstrate the superiority of the proposed SHCBR performance, we compare it with the following models:

- MFBPR [15]: This is a matrix factorization model based on BPR loss optimization, which is widely used for implicit feedback.
- RGCN [39]: RGCN is a method based on graph convolutional networks that is specifically designed to handle multi-relational graphs.
- LightGCN [13]: This is a recommendation model based on graph neural network and collaborative filtering ideas, which is lightweight and efficient.
- BundleNet [5]: BundleNet constructs a user-bundle-item tripartite graph. It utilizes graph neural networks to learn node representations while performing multi-task learning.
- DAM [17]: DAM is a deep learning model which combines attention mechanisms and multi-task learning framework to learn the bundle representations.
- BGCN [18]: BGCN is a bundle recommendation model based on graph neural networks. It leverages the powerful ability of graph neural networks in learning from complicated topology and higher-order connectivity, effectively modeling the complex relations between users, items, and bundles.
- MIDGN [20]: MIDGN models the user's intent under global and local views and adopts the contrast learning framework, disentangling the user and bundle representations at the granularity of the user's intents.

5.1.3. Hyper-parameter settings

For all methods, we fix the embedding size to 128. For SHCBR, we implement it using the PyTorch framework(<https://pytorch.org/>) and use AdamW optimizer for model optimization. We use a batch size of 2048 for both datasets in our experiments. Then, We search the learning rate in {1e-4, 5e-4, 1e-3, 5e-3, 1e-2}. The final experimental results verify that the best choice is 5e-3. In addition, we adopt the L2-norm with 0.2 and dropout with 0.2 to prevent overfitting.

5.2. Performance Comparison and Analysis

We first compare the overall recommendation performance of SHCBR with the existing baselines on two datasets. As shown in Table 2, SHCBR achieves the best performance. We highlight the best-performing method in bold and underline the strongest performing baseline. From the Table 2, we make the following observations:

Table 2. The performance of SHCBR with other baseline methods over two datasets.

Method	NetEase				Youshu			
	Recall @20	NDCG @20	Recall @40	NDCG @40	Recall @20	NDCG @20	Recall @40	NDCG @40
MFBPR	0.0355	0.0181	0.0600	0.0246	0.1959	0.1117	0.2735	0.1320
RGCN	0.0407	0.0210	0.0670	0.0280	0.2040	0.1069	0.3017	0.1330
LightGCN	0.0496	0.0254	0.0795	0.0334	0.2286	0.1344	0.3190	0.1592
BundleNet	0.0391	0.0201	0.0661	0.0271	0.1895	0.1125	0.2675	0.1335
DAM	0.0411	0.0210	0.0690	0.0281	0.2082	0.1198	0.2890	0.1418
BGCN	0.0491	0.0258	0.0829	0.0346	0.2347	0.1345	0.3248	0.1593
MIDGN	<u>0.0678</u>	<u>0.0343</u>	<u>0.1085</u>	<u>0.0451</u>	<u>0.2682</u>	<u>0.1527</u>	<u>0.3712</u>	<u>0.1808</u>
Ours	0.0852	0.0458	0.1304	0.0561	0.3049	0.1838	0.4123	0.2112
Imp%	25.66%	33.53%	20.18%	24.39%	13.68%	20.37%	11.07%	16.81%

Compared with the traditional machine learning method BPR, the graph-based methods LightGCN and RGCN exhibit more powerful learning capabilities. This is attributed to the advantages of graph neural networks in capturing graph structures and aggregating multi-hop collaborative

information. The performance of RGCN demonstrates the importance of modeling relationship between entities in personalized bundle recommendation tasks. Among the three methods, LightGCN performs the best, thanks to its simplified design of GCN, making it more suitable for bundle recommendation scenarios.

In recent years, significant progress has been made in research on bundle recommendation. It is worth mentioning that BundleNet constructs a user-item-bundle tripartite graph and formalize the bundle recommendation task as a link prediction problem on the graph. However, its performance is not satisfactory due to its overly simplistic neural network model. It has also been observed that some excellent graph neural network-based models (such as RGCN and BundleNet) perform even worse than non-graph machine learning methods like DAM, which leverages deep attention mechanisms and a multi-task framework to capture user preferences jointly. In addition, BGCN, based on a graph neural network, explicitly model complex relations between users, items, and bundles to solve the problem of bundle recommendation effectively. MIDGN decouples user-item and bundle-item interactions to capture the potential intentions of users and bundles from different perspectives. At the same time, it is also the best model among the bundle recommendation methods in the baseline. While graph-based methods are effective, hypergraphs offer a more flexible structure. Capturing complex associations between users, items, and bundles is crucial in bundle recommendation tasks, and hypergraphs have a natural advantage in handling higher-order associations. Therefore, our proposed SHCBR achieves the best results. As shown in Table 2, SHCBR significantly outperforms all baselines in terms of Recall and NDCG metrics. Specifically, SHCBR outperforms the best baselines by 20.18%-33.53% on the NetEase dataset. The performance of SHCBR on Youshu dataset is 11.07%-20.37% higher than the best baseline.

5.3. Ablation Study of SHCBR

Next, we conduct ablation studies to analysis the effectiveness of several key designs in SHCBR. We sequentially remove the core components of SHCBR to form different derivative models. Referring to the results displayed in Table 3, we have the following observations:

- $SHCBR_{w/oHG}$. This model removes the module of the relational hypergraph construction. In this part, we exclude the similarity overlap matrix form the user-bundle adjacency matrix. At the same time, we also eliminate the construction of the structural hypergraph matrix. It can be observed that SHCBR performs significantly better than $SHCBR_{w/oHG}$, demonstrating the effectiveness of the relational hypergraph construction module. In addition, the experimental results also highlight the importance of the hypergraph structure in capturing node feature.
- $SHCBR_{w/oHC}$. This model removes the part of the structural hypergraph convolution. Here, we replace the structural hypergraph convolutional neural network with a simple graph convolutional neural network. We can find that the performance of SHCBR is better than $SHCBR_{w/oHC}$. This is due to the insufficient aggregation capability of simple graph convolutional neural networks compared to hypergraph convolutional neural networks. This demonstrates the superiority of our proposed structural hypergraph convolutional neural network.
- $SHCBR_{w/oMP}$. This model removes the special matrix propagation rule module but retains other designs of SHCBR. It can be observed that SHCBR is only slightly superior than $SHCBR_{w/oMP}$. Although SHCBR is not highly competitive compared to $SHCBR_{w/oHG}$, we can still see that the special matrix propagation rule is helpful for improving model performance.

Table 3. Ablation Study

Method	NetEase		Youshu	
	Recall@20	NDCG@20	Recall@20	NDCG@20
SHCBR_w/o HG	0.0801	0.0397	0.2879	0.1743
SHCBR_w/o HC	0.0797	0.0411	0.2838	0.1716
SHCBR_w/o MP	0.0832	0.0441	0.2978	0.1806
SHCBR	0.0852	0.0458	0.3049	0.1838

5.4. Hyper-parameters Analysis

We further investigate the impact of hyper-parameters α and batch size on the performance of SHCBR.

Research on hyper-parameters α As shown in Figure 3, we analyze the ratio of positive to negative sample' loss function in Equation 15 to explore the impact of this ratio on the performance of SHCBR. We compare the SHCBR's Recall@20 and NDCG@20 when setting different α . We can observe that for both datasets, the highest NDCG@20 is achieved when the value of α is 0.4, and the highest Recall@20 is achieved when the value of α is 0.3. Therefore, we can conclude that the influence of positive samples is slightly greater than that of negative samples. This situation arises mainly due to the random selection strategy for negative samples.

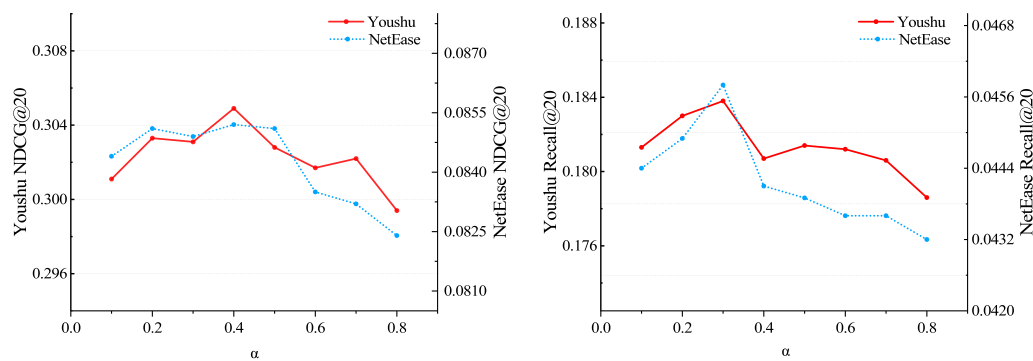


Figure 3. Impact of hyper-parameters α .

Research on batch size To study the impact of batch size on the model, we gradually increase the batch size from 128 to 8192. As shown in Figure 4, we can observe that as the batch size increases, the performance of the model initially improves rapidly and then reaches a stable state. Based on the observed results, we use a batch size of 2048 in our experiments on both datasets.

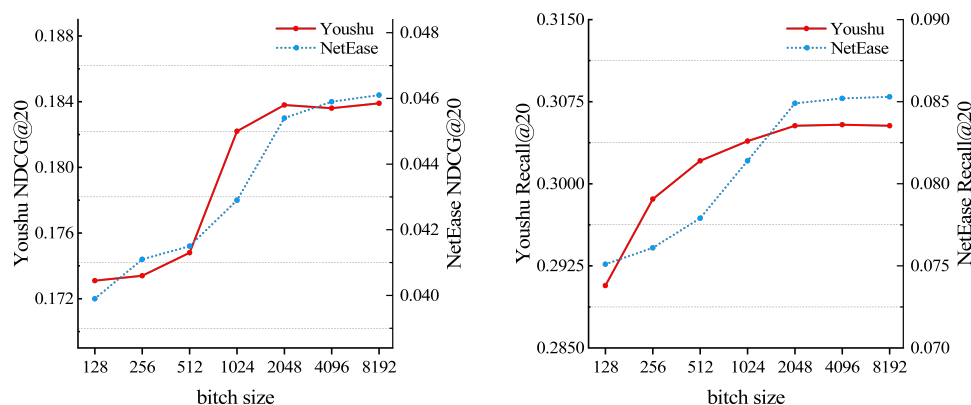


Figure 4. Impact of Bitch Size.

6. Conclusion

In this work, we investigate the bundle recommendation task. Unlike traditional single-item recommendation tasks, bundle recommendation involves recommending a set of related items, i.e., bundles, to users. We propose a novel model named SHCBR, which jointly incorporates user nodes, item nodes and bundle nodes into a relational hypergraph from a global perspective. We

utilize flexible hypergraph structure to model multiple complex associations among users, items, and bundles. With item nodes as links, we leverage efficient hypergraph convolution to learn the hidden layer representation considering the high-order information, which improves the quality of node representation. This modeling approach allows for better exploration of the underlying interests and associations behind user behavior, alleviating the dilemma of data scarcity. Extensive experiments demonstrate that our proposed SHCBR outperforms the baseline methods on two real-world datasets. In the future, we will investigate the use of the contrast learning framework to improve recommendation efficiency while addressing data sparsity.

Author Contributions: X.L.: conceptualization, methodology, software, investigation, writing—original draft preparation, writing—review and editing; M.Y.: supervision, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Heilongjiang Provincial Philosophical and Social Science Research Planning Project of China(No.19EDE334).

Data Availability Statement: The experimental codes and datasets are available at <https://github.com/Lxtdzh/SHCBR-master/>. (accessed on 2 August 2023).

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

References

1. Wu, S.; Sun, F.; Zhang, W.; Xie, X.; Cui, B. Graph neural networks in recommender systems: a survey. *Acm Computing Surveys* **2022**, *55*, 1–37.
2. Covington, P.; Adams, J.; Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*; 2016; pp. 191–198.
3. Cao, D.; Nie, L.; He, X.; Wei, X.; Zhu, S.; Chua, T. Embedding factorization models for jointly recommending items and user generated lists. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*; pp. 585–594.
4. Pathak, A.; Gupta, K.; McAuley, J. Generating and personalizing bundle recommendations on steam. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*; 2017; pp. 1073–1076.
5. Deng, Q.; Wang, K.; Zhao, M.; Zou, Z.; Wu, R.; Tao, J.; Fan, C.; Chen, L. Personalized bundle recommendation in online games. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*; 2020; pp. 2381–2388.
6. Zheng, Z.; Wang, C.; Xu, T.; Shen, D.; Qin, P.; Huai, B.; Liu, T.; Chen, E. Drug package recommendation via interaction-aware graph induction. In *Proceedings of the Web Conference 2021*; 2021; pp. 1284–1295.
7. Zhu, T.; Harrington, P.; Li, J.; Tang, L. Bundle recommendation in ecommerce. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*; 2014; pp. 657–666.
8. Su, Y.; Zhang, R.; Erfani, S.; Xu, Z. Detecting beneficial feature interactions for recommender systems. In *Proceedings of the AAAI conference on artificial intelligence*; 2021; pp. 4357–4365.
9. Scarselli, F.; Gori, M.; Tsoi, A.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Transactions On Neural Networks* **2008**, *20*, 61–80.
10. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81.
11. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S. A comprehensive survey on graph neural networks. *IEEE Transactions On Neural Networks And Learning Systems* **2020**, *32*, 4–24.
12. Bretto, A. Hypergraph theory. *An Introduction. Mathematical Engineering*. Cham: Springer **2013**, *1*.
13. 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*; 2020; pp. 639–648.
14. Liu, Y.; Xie, M.; Lakshmanan, L. Recommending user generated item lists. In *Proceedings of the 8th ACM Conference on Recommender systems*; 2014; pp. 185–192.
15. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *Arxiv Preprint Arxiv:1205.2618* **2012**.

16. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. ; Dean, J. Distributed representations of words and phrases and their compositionality *Advances In Neural Information Processing Systems* **2013**, 26.
17. Chen, L.; Liu, Y.; He, X.; Gao, L. ; Zheng, Z. Matching user with item set: Collaborative bundle recommendation with deep attention network. In *IJCAI*; 2019; pp. 2095–2101.
18. Chang, J.; Gao, C.; He, X.; Jin, D. ; Li, Y. Bundle recommendation and generation with graph neural networks *IEEE Transactions On Knowledge And Data Engineering* **2021**, 35, 2326–2340.
19. Kipf, T. ; Welling, M. Semi-supervised classification with graph convolutional networks. *Arxiv Preprint Arxiv:1609.02907* **2016**.
20. Zhao, S.; Wei, W.; Zou, D. ; Mao, X. Multi-view intent disentangle graph networks for bundle recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*; 2022; pp. 4379–4387.
21. Sun, J.; Wang, N. ; Liu, X. IMBR: Interactive Multi-relation Bundle Recommendation with Graph Neural Network. In *International Conference on Wireless Algorithms, Systems, and Applications*; 2022; pp. 460–472.
22. Yadati, N.; Nitin, V.; Nimishakavi, M.; Yadav, P.; Louis, A. ; Talukdar, P. NHP: Neural hypergraph link prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*; 2020; pp. 1705–1714.
23. Xu, Y.; Zhang, H.; Cheng, K.; Liao, X.; Zhang, Z. ; Li, Y. Knowledge graph embedding with entity attributes using hypergraph neural networks. *Intelligent Data Analysis* **2022**, 26, 959–975.
24. Gao, Y.; Zhang, Z.; Lin, H.; Zhao, X.; Du, S. ; Zou, C. Hypergraph learning: Methods and practices. *IEEE Transactions On Pattern Analysis And Machine Intelligence* **2020**, 44, 2548–2566.
25. Zhou, D.; Huang, J. ; Schölkopf, B. Learning with hypergraphs: Clustering, classification, and embedding. *Advances In Neural Information Processing Systems* **2006**, 19.
26. Huang, Y.; Liu, Q. ; Metaxas, D. Video object segmentation by hypergraph cut. In *2009 IEEE conference on computer vision and pattern recognition*; 2009; pp. 1738–1745.
27. Gao, Y.; Wang, M.; Tao, D.; Ji, R. ; Dai, Q. 3-D object retrieval and recognition with hypergraph analysis. *IEEE Transactions On Image Processing* **2012**, 21, 4290–4303.
28. Gao, Y.; Wang, M.; Zha, Z.; Shen, J.; Li, X. ; Wu, X. Visual-textual joint relevance learning for tag-based social image search. *IEEE Transactions On Image Processing* **2012**, 22, 363–376.
29. Feng, Y.; You, H.; Zhang, Z.; Ji, R. ; Gao, Y. Hypergraph neural networks. In *2009 IEEE conference on computer vision and pattern recognition*; 2019; pp. 3558–3565.
30. Bai, S.; Zhang, F. ; Torr, P. Hypergraph convolution and hypergraph attention. *Pattern Recognition* **2021**, 110 pp. 107637.
31. Wang, M.; Liu, X. ; Wu, X. Visual classification by ℓ_1 -hypergraph modeling. *IEEE Transactions On Knowledge And Data Engineering* **2015**, 27, 2564–2574.
32. Author 1, T. The title of the cited article. *Journal Abbreviation* **2008**, 10, 142–149. Yu, Z.; Li, J.; Chen, L. ; Zheng, Z. Unifying multi-associations through hypergraph for bundle recommendation. *Knowledge-based Systems* **2022**, 255, pp. 109755.
33. Wang, Z.; Chen, J.; Rosas, F. ; Zhu, T. A hypergraph-based framework for personalized recommendations via user preference and dynamics clustering. *Expert Systems With Applications* **2022**, 204 pp. 117552.
34. Author 1, T. The title of the cited article. *Journal Abbreviation* **2008**, 10, 142–149. Peng, D. ; Zhang, S. GC-HGNN: A global-context supported hypergraph neural network for enhancing session-based recommendation. *Electronic Commerce Research And Applications* **2022**, 52 pp. 101129.
35. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P. ; Bengio, Y. Graph attention networks. *Arxiv Preprint Arxiv:1710.10903* **2017**.
36. Maas, A.; Hannun, A. ; Ng, A. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*; 2013; pp. 3.
37. Clevert, D.; Unterthiner, T. ; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv 2015. Arxiv Preprint Arxiv:1511.07289* **2016**, 2.

38. Loshchilov, I. ; Hutter, F. Decoupled weight decay regularization. *Arxiv Preprint Arxiv:1711.05101* **2017**.
39. Wang, X.; Liu, X.; Liu, J. ; Wu, H. Relational graph neural network with neighbor interactions for bundle recommendation service. In *2021 IEEE International Conference on Web Services (ICWS)*; 2021; pp. 167–172.

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.