

Multi-View Hybrid Contrastive Learning for Bundle Recommendation

Maoyan Lin, Youxin Hu, Zhixin Wang, Jianqiu Luo, Jinyu Huang

School of Computer Science and Engineering, Sichuan University of Science and Engineering, Yibin, China

Email: miai1590811@gmail.com

How to cite this paper: Lin, M.Y., Hu, Y.X., Wang, Z.X., Luo, J.Q. and Huang, J.Y. (2023) Multi-View Hybrid Contrastive Learning for Bundle Recommendation. *Open Journal of Applied Sciences*, 13, 1742-1763.

<https://doi.org/10.4236/ojapps.2023.1310138>

Received: September 27, 2023

Accepted: October 22, 2023

Published: October 25, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Bundle recommendation aims to provide users with convenient one-stop solutions by recommending bundles of related items that cater to their diverse needs. However, previous research has neglected the interaction between bundle and item views and relied on simplistic methods for predicting user-bundle relationships. To address this limitation, we propose Hybrid Contrastive Learning for Bundle Recommendation (HCLBR). Our approach integrates unsupervised and supervised contrastive learning to enrich user and bundle representations, promoting diversity. By leveraging interconnected views of user-item and user-bundle nodes, HCLBR enhances representation learning for robust recommendations. Evaluation on four public datasets demonstrates the superior performance of HCLBR over state-of-the-art baselines. Our findings highlight the significance of leveraging contrastive learning and interconnected views in bundle recommendation, providing valuable insights for marketing strategies and recommendation system design.

Keywords

Recommender Systems, Bundle Recommendation, Package Recommendation, Contrastive Learning, Graph Neural Network

1. Introduction

In e-commerce, product bundle sales are an important marketing strategy to support promotional activities, attract customers, and increase sales revenue [1] [2] [3] [4]. It involves grouping together a collection of related products that users tend to consume as a whole in certain situations, such as at a discounted total price [5] [6] [7], or for a specific purpose [8] [9]. For illustrative purposes, **Figure 1** depicts three example bundles in the fields of clothing, electronics, and



Figure 1. Electronics, books, and clothing bundle from amazon.

books from Amazon¹. The bundle on the far right of the image shows a set of clothing items suitable for the spring season, including glasses, a hoodie, shorts, and sneakers. Such a bundle may be appealing to customers who are uncertain about what to wear in the spring and are looking for a well-coordinated set of fashionable products, rather than unrelated items.

Most general recommender systems generate recommended items for users based on their preferences and extracted item features from user-item interaction history [10]. Intuitively, we can consider bundles as virtual items and apply a traditional recommendation algorithm for items. However, the lack of user-bundle interactions can lead to poor performance in bundle recommendations [11]. Bundle recommendation is also a challenging task as it is difficult to capture the intricate relationships between items, bundles, and users. Many previous recommendation algorithms typically focus on user preferences for bundles and bundle features, disregarding the correlations between items and bundles, as well as the fact that user preferences for bundles and items can be further leveraged to improve recommendation performance [12].

Recently, many bundle recommendation methods were proposed such as BGCN [13], DAM [11], and BundleNet [14]. DAM and BundleNet advance bundle recommendation through a multi-task learning framework, using the Bayesian personalized ranking (BPR) loss function [12], and enhancing bundle recommendation performance by jointly predicting items and bundles. However, these methods do not effectively distinguish between user preferences for items versus user preferences for bundles, and overlook the mutual influence between user preferences for items and bundles. While BGCN has made significant contributions by separately modeling user preferences for items and bundles, it still neglects the mutual influence between user preferences for items and bundles, which should be derived from the association between items and bundles. While CrossCBR [15] provides a partial solution to address the above problems, it overlooks the potential complementary relationships among multiple views and the unsupervised nature of contrastive learning. Taking the example in **Figure 2** into consideration, where user u_1 is the target user who has interacted with bundle b_1 and items i_1 , i_2 , and i_4 , it is likely to recommend

¹www.amazon.com

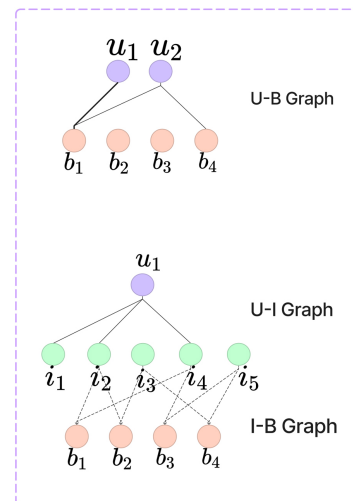


Figure 2. The bundle and item views are depicted in the U-B, U-I, and I-B graphs.

bundle b_4 to user u_1 if we can effectively capture the similarity in behavior between user u_1 and user u_2 . On the other hand, from the item view, the recommendation model tends to recommend bundles b_2 and b_3 as interesting bundles for user u_1 . This is because bundle b_2 contains items that are common with historical bundle b_1 (*i.e.*, i_2), and bundle b_3 contains items that are independently preferred by user u_1 (*i.e.*, i_4). Clearly, the bundle view emphasizes the similarity in behavior among users, while the item view emphasizes the content relevance among bundles and user-level preferences for items. Therefore, they are complementary but different, and the collaborative connection between these two views is crucial for accurate bundle recommendations.

In the article, we propose a multi-view Hybrid Contrastive Learning for Bundle Recommendation (HCLBR), which captures collaborative associations through multi-view contrastive learning and enhances perspective-aware representations through mutual reinforcement. The basic idea is to treat bundle and item views as two independent but interrelated perspectives of user-bundle preference and transform their consistency into representations of users and bundles through contrastive learning on these views. The contributions of our work are summarized as follows:

- We introduced an important concept, which is to establish a collaborative relationship between the bundle view and the item view, playing a crucial role in addressing the bundle recommendation problem.
- We proposed the Hybrid Contrastive Learning-based Bundle Recommendation (HCLBR) model, which to the best of our knowledge, is the first work to jointly leverage unsupervised and supervised contrastive learning for bundle recommendation.
- Our model demonstrates superior performance compared to state-of-the-art baselines on four publicly available datasets, while also substantially reducing training time.

2. Related Work

2.1. Graph-Based Recommendation

In recent years, graph-based models have gained prominence as the primary approach for collaborative filtering in recommender systems. This is attributed to their ability to effectively capture the intricate interaction patterns between users and items [16] [17] [18] [19]. Among these models, those based on graph neural networks, such as NGCF [18] and LightGCN [17], have emerged as popular research directions. The NGCF model constructs a bipartite graph using the user-item interaction matrix and leverages graph convolutional networks for effective graph learning. On the other hand, LightGCN simplifies the NGCF model by removing non-linear feature transformations and activation function layers, resulting in improved performance. Notably, LightGCN has demonstrated outstanding performance across various recommendation tasks [20].

2.2. Bundle Recommender Systems

In recent years, there has been a rising interest in the study of bundle recommendation. Among these approaches, the EFM [21] model utilizes embedding-based techniques to uncover associations between items and item lists. The DAM [11] model introduces a decomposed attention network to aggregate project information and jointly models user bundle interactions and user-item interactions. Graph-based learning for recommendation systems has also gained attention as a promising field. For example, BGCN [13] constructs a heterogeneous graph consisting of user, item, and bundle nodes, enabling the learning of latent factors while propagating interaction and association information. BRUCE [22] pioneers the use of Transformers to model user bundle preferences and relationships among items within bundles. CrossCBR employs cross-view contrastive learning to simulate cross-view collaborative associations in bundle recommendation. The BundleNet [14] model constructs a user-item-bundle tripartite graph from historical interactions and extends it to a relational graph using the GCN model. However, BundleNet confuses the relationships between users, bundles, and items, whereas BGCN decomposes user preferences into item view and bundle view, effectively capturing preferences for both types and achieving superior performance.

2.3. Contrastive Learning in Recommendation

A wave of attention towards adaptive learning has been directed towards recommendation tasks to address the issue of sparse labels. SGL [23] generates multiple views for nodes and maximizes the consistency between different views of the same node. CCGL [24] and HeCo [25] are two variants used on cascade graphs or heterogeneous graphs, respectively. SSL4Rec [26] applies data augmentation on item features and introduces a contrastive pretraining objective to enhance learning representations in the Siamese model. In the domain of knowledge-aware recommendation, Yang *et al.* [27] developed a knowledge graph

contrastive learning framework called KGCL to facilitate denoising and integration between CF learning and knowledge graph encoding. For social-aware recommendation, Fan *et al.* [28] introduced a novel graph neural network framework called GraphRec. In the field of sequential recommendation, Xie *et al.* [29] introduced sequence data augmentation into contrastive learning tasks to obtain more robust sequential representations. Qiu *et al.* [30] proposed a contrastive learning method based on sequence-level positive pairs to address the issue of representation degradation in sequential recommendation systems. COTREC [31] constructs two views, namely the item view and session view, to learn session representations from two data sources: the session-to-item transition graph and the session-session similarity graph. Then contrastive learning is applied based on these perspectives. ICLRec [32] performs user intent clustering and contrastive learning to enhance sequential recommendation by improving the representation of user interests.

3. Methodology

3.1. Problem Formulation

Given a set of users $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$, a set of bundles $\mathcal{B} = \{b_1, b_2, \dots, b_L\}$, and a set of items $\mathcal{I} = \{i_1, i_2, \dots, i_N\}$, where M , L , and N represent the number of users, bundles, and items, respectively. The user-bundle interaction, user-item interaction, and bundle-item affiliation are represented as

$X_{M \times L} = \{x_{ub} \mid u \in \mathcal{U}, b \in \mathcal{B}\}$, $Y_{M \times N} = \{y_{ui} \mid u \in \mathcal{U}, i \in \mathcal{I}\}$, and $Z_{L \times N} = \{z_{bi} \mid b \in \mathcal{B}, i \in \mathcal{I}\}$, respectively, $x_{ub}, y_{ui}, z_{bi} \in \{0, 1\}$. In this context, 1 denotes a connection between the user and a bundle or item, or signifies the inclusion of the item within a specific bundle. Please note that due to duplicate data elimination in the user's historical bundle and item interactions, each element of X and Y is represented as a binary value instead of an integer. Additionally, X and Y are generated separately, allowing users to directly interact with bundles and individual items. As a result, X and Y contain different information, which heuristically facilitates collaboration between two distinct views. The objective of the bundle recommendation task is to learn from the historical $\{X, Y, Z\}$ data and predict user-bundle interactions that are unseen in X .

3.2. Learning Representations from Multiple Views

For the first component of HCLBR, our goal is to learn representations from multiple views, namely the original and modified views of items and bundles. While the dual-view representation learning module of BGCN is effective, its design for graph construction and graph learning is deemed useless or even detrimental, especially when contrastive learning is employed [17]. Here, we propose a simpler yet more effective approach for representation learning.

3.2.1. Bundle-View Representation Learning

To learn user and bundle representations from the bundle view, we first con-

struct a user-bundle bipartite graph, known as the U-B graph, using the user-bundle interaction matrix X . Next, we employ the widely used GNN-based recommendation framework called LightGCN to learn representations for users and bundles. In this framework, we conduct information propagation on the U-B graph, where the k -th layer's information propagation is represented as follows:

$$\begin{cases} e_u^{B(k)} = \sum_{b \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|} \sqrt{|\mathcal{N}_b|}} e_b^{B(k-1)}, \\ e_b^{B(k)} = \sum_{u \in \mathcal{N}_b} \frac{1}{\sqrt{|\mathcal{N}_b|} \sqrt{|\mathcal{N}_u|}} e_u^{B(k-1)}, \end{cases} \quad (1)$$

where $e_u^{B(k)}, e_b^{B(k)} \in \mathbb{R}^d$ represent the information propagation for user u and bundle b at the k -th layer. Here, d represents the embedding dimension, indicating the size of the representation vectors. The superscript B indicates that the propagation is performed on the bundle view. Initially, $e_u^{B(0)}$ and $e_b^{B(0)}$ are randomly initialized at the start of the training process. \mathcal{N}_u and \mathcal{N}_b denote the neighbors of user u and bundle b , respectively, in the U-B graph.

We adopt the methodology of LightGCN by removing self-connections from the U-B graph and excluding non-linear transformations from the information propagation function. Through empirical validation, we have confirmed that this simplification, which was not considered in BGCN, indeed contributes to performance improvements. Importantly, we do not include bundle-bundle connections introduced by BGCN, which are calculated based on the degree of overlap between two bundles in terms of shared items. This is because the information regarding bundle-bundle overlap can be effectively extracted through the utilization of multi-view contrastive learning from the item view. Moreover, removing the additional bundle-bundle connections can further reduce the computational costs associated with graph learning.

We connect the embeddings from all K layers to aggregate the information received from neighbors at different depths. The final representations of the bundle view, which are denoted as e_u^{B*} and e_b^{B*} , can be expressed as:

$$e_u^{B*} = \sum_{k=0}^K e_u^{B(k)}, \quad e_b^{B*} = \sum_{k=0}^K e_b^{B(k)} \quad (2)$$

3.2.2. Item-View Representation Learning

To learn user and bundle representations from the item view, we start by constructing two bipartite graphs: the U-I (user-item) graph and the B-I (bundle-item) graph. Just like in the U-B graph learning process, we employ the LightGCN framework to learn representations for users and items. The resulting user representations are known as item-view user representations. On the other hand, the item-view bundle representations are obtained by performing average pooling on the item representations, guided by the B-I graph. Specifically, the information propagation on the U-I graph is defined as follows:

$$\begin{cases} e_u^{I(k)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|} \sqrt{|\mathcal{N}_i|}} e_i^{I(k-1)}, \\ e_i^{I(k)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|} \sqrt{|\mathcal{N}_u|}} e_u^{I(k-1)} \end{cases} \quad (3)$$

where $e_u^{I(k)}, e_i^{I(k)} \in \mathbb{R}^d$ represent the information propagated to user u and item i at the k -th layer, respectively. The superscript I indicates the item view. The initial representation $e_i^{I(0)}$ is randomly initialized. \mathcal{N}_u and \mathcal{N}_i denote the neighbors of users and items in the U-I graph, respectively. We follow the approach of BGCN and share the parameters of $e_u^{I(0)}$ with $e_u^{B(0)}$. Empirically, it has been observed that this parameter sharing does not affect performance and greatly reduces the parameter count. Similar to the U-B graph, we remove self-connections from the U-I graph and exclude non-linear feature transformations from the information propagation function. Additionally, we incorporate a layer aggregation operation after K layers of information propagation. The specific formula for this operation is as follows:

$$e_u^{I*} = \sum_{k=0}^K e_u^{I(k)}, \quad e_i^{I*} = \sum_{k=0}^K e_i^{I(k)}, \quad (4)$$

where e_u^{I*} and e_i^{I*} represent the user and item representations in the item-view, respectively. Based on the item view item representations and the B-I graph, we can obtain the item-view bundle representation e_b^{I*} through average pooling, expressed as:

$$e_b^{I*} = \frac{1}{|\mathcal{N}_b|} \sum_{i \in \mathcal{N}_b} e_i^{I*}, \quad (5)$$

where \mathcal{N}_b represents the set of items contained in a certain bundle b .

In conclusion, we can learn representations for all users and bundles from two views, denoted as $E_U^{B*}, E_U^{I*} \in \mathbb{R}^{M \times d}$ and $E_B^{B*}, E_B^{I*} \in \mathbb{R}^{L \times d}$, where the superscripts B and I indicate the bundle and item views, respectively. And the subscripts U and B represent the user set and bundle set, respectively. Additionally, $E_I^{I*} \in \mathbb{R}^{N \times d}$ represents the item representations in the item view. Then, given a user and a bundle, we can obtain their bundle-view representations *i.e.*, e_u^{B*} and e_b^{B*} , and their item-view representations *i.e.*, e_u^{I*} and e_b^{I*} .

3.3. Joint Optimization

We have devised crucial components to model cooperative relationships across multiple views using hybrid contrastive learning. Firstly, we introduce the commonly employed Bayesian Personalized Ranking (BPR) in the supervised learning paradigm. Subsequently, we present the data augmentation method and provide an overview of contrastive loss. Lastly, we discuss the process of joint optimization.

3.3.1. Bayesian Personalized Ranking Loss

The probability of user u adopting bundled package b is estimated by calculating the inner product of the user vector e_u and the item vector e_b . In the super-

vised learning paradigm, a common practice is to employ the Bayesian Personalized Ranking (BPR) loss, which assigns higher probabilities to observed interactions than unobserved interactions. To obtain the final recommendation prediction, we first compute predictions for the item view and bundled view using inner products, and then combine them additively for the ultimate prediction.

$$y_{u,b}^* = e_u^{B*\top} e_b^{B*} + e_u^{I*\top} e_b^{I*}. \quad (6)$$

The conventional Bayesian Personalized Ranking loss is utilized as the primary loss function.

$$\mathcal{L}_{BPR} = \sum_{(u,b,b^*) \in Q} -\ln s(y_{u,b}^* - y_{u,b^*}^*). \quad (7)$$

where $Q = \{(u, b, b^*) \mid u \in U, b, b^* \in \mathcal{B}, x_{ub} = 1, x_{ub^*} = 0\}$, and $\sigma(\cdot)$ denotes the sigmoid function.

3.3.2. Data Augmentation

The fundamental idea behind self-supervised contrastive learning is to foster affinity among different perspectives of the same object while simultaneously expanding the diversity of representations across distinct objects [33]. In practical terms, each object typically possesses multiple natural views, such as images captured from different angles or bundled and itemized views in recommendation systems, where contrastive loss can be directly applied. However, there are situations where multiple views are unavailable, necessitating the use of data augmentation techniques to generate additional perspectives from the original data [23] [34] [35]. By employing appropriate data augmentation, we not only unlock the potential of applying contrastive learning to multi-view data but also bolster the robustness against potential adversarial noise. Therefore, while keeping the original data (without augmentation) as the default configuration, we also introduce two straightforward data augmentation methods: graph-based augmentation and embedding-based augmentation.

Graph-Based Augmentation. The primary objective of graph augmentation is to generate augmented data by modifying the graph structure [23]. We employ a straightforward approach called Edge Dropout for random augmentation, where a certain proportion (dropout rate ρ) of edges is randomly removed from the original graph. The underlying principle behind edge dropout is to preserve the essential local structure of the graph, thereby enhancing the robustness of learning representations against certain types of noise.

Embedding-Based Augmentation. In contrast to graph-based augmentation, which is limited to graph data, embedding-based augmentation is more versatile and can be applied to any deep representation learning methods [35]. The core concept is to alter the learned representation embeddings, irrespective of their acquisition process. To achieve this, we employ Message Dropout, which randomly masks certain elements of the propagated embeddings with a dropout rate ρ during the graph learning process.

Raw Preservation. We refer to the method without any data augmentation as

raw preservation, where no randomness is introduced, and the original representations are solely retained. As the two views in bundled recommendation are derived from distinct data sources, their representations possess ample diversity, leading to favorable outcomes.

To avoid notational confusion, we denote the augmented representations of all users and bundles as $E_{U'}^{B*}, E_{U'}^{I*} \in \mathbb{R}^{M' \times d}$ and $E_{B'}^{B*}, E_{B'}^{I*} \in \mathbb{R}^{L' \times d}$. Here, the superscripts B and I represent the bundle and item views, respectively, while the subscripts U' and B' indicate the entire set of users and the bundle set, respectively ($E_{I'}^{I*} \in \mathbb{R}^{N' \times d}$ represents the representation of all items in the item view). Specifically, $\mathcal{U}' = \{u_1, u_2, \dots, u_{M'}\}$, $\mathcal{B}' = \{b_1, b_2, \dots, b_{L'}\}$ and $\mathcal{I}' = \{i_1, i_2, \dots, i_{N'}\}$. Subsequently, when considering a specific user and bundle, we can obtain their augmented representations in the bundle view, denoted as *i.e.*, $e_{u'}^{B*}$ and $e_{b'}^{B*}$, as well as their representations in the item view, such as *i.e.*, $e_{u'}^{I*}$ and $e_{b'}^{I*}$.

3.3.3. Hybrid Contrastive Learning

Unsupervised Contrastive Learning. Recent research has explored the application of contrastive learning on graphs to address the challenge of sparse labels and enhance model robustness [23]. Given a pair of generated graph views [36], such as SGL, it proposes to bring together different views of the same node while separating views of different nodes. As depicted in Figure 3, each view captures a distinct aspect of user preferences, and these two views must synergistically work together to maximize the overall modeling capability. The InfoNCE loss [37] is employed to compute the unsupervised contrastive learning (UCL) loss, which involves node discrimination and is expressed by the following equation:

$$\mathcal{L}_{UCL}(U^{B*}, U^{I*}) = \sum_{u' \in \mathcal{U}'} -\log \frac{\exp(f(e_{u'}^{B*}, e_{u'}^{I*})/\tau)}{\sum_{v' \in \mathcal{U}'} \exp(f(e_{u'}^{B*}, e_{v'}^{I*})/\tau)} \quad (8)$$

where, $f(\dots)$ denotes the cosine similarity function, and τ represents a temperature hyper-parameter. The different perspectives of the same user are considered as positive pairs $(e_{u'}^{B*}, e_{u'}^{I*})$, encouraging consistent behavior across these pairs. Conversely, viewpoints from different users are treated as negative pairs $(e_{u'}^{B*}, e_{v'}^{I*})$, aiming to minimize their mutual agreement. Similarly, we derive the UCL loss $\mathcal{L}_{UCL}(B^{B*}, B^{I*})$ for the bundle side using a similar approach.

Supervised Contrastive Learning. While unsupervised contrastive learning has shown significant improvements, it overlooks the usefulness of available user-bundle interactions when dealing with different graph views. In this context, we propose a hybrid contrastive learning module that effectively utilizes the available user-bundle interactions. As illustrated in Figure 3, besides unsupervised contrastive learning for $(e_{u'}^{B*}, e_{u'}^{I*})$ and $(e_{u'}^{B*}, e_{b'}^{I*})$, we suggest promoting consistency in the presence of observed user-bundle interactions $(e_{u'}^{I*}, e_{b'}^{B*})$ by computing a supervised contrastive learning (SCL) loss for the embeddings of users and interacting bundles. Following the same underlying principle of contrastive learning, on one hand, given the observed user-bundle interaction y_{ub} , we aim to maximize the consistency between the user representation $e_{u'}^{I*}$ and

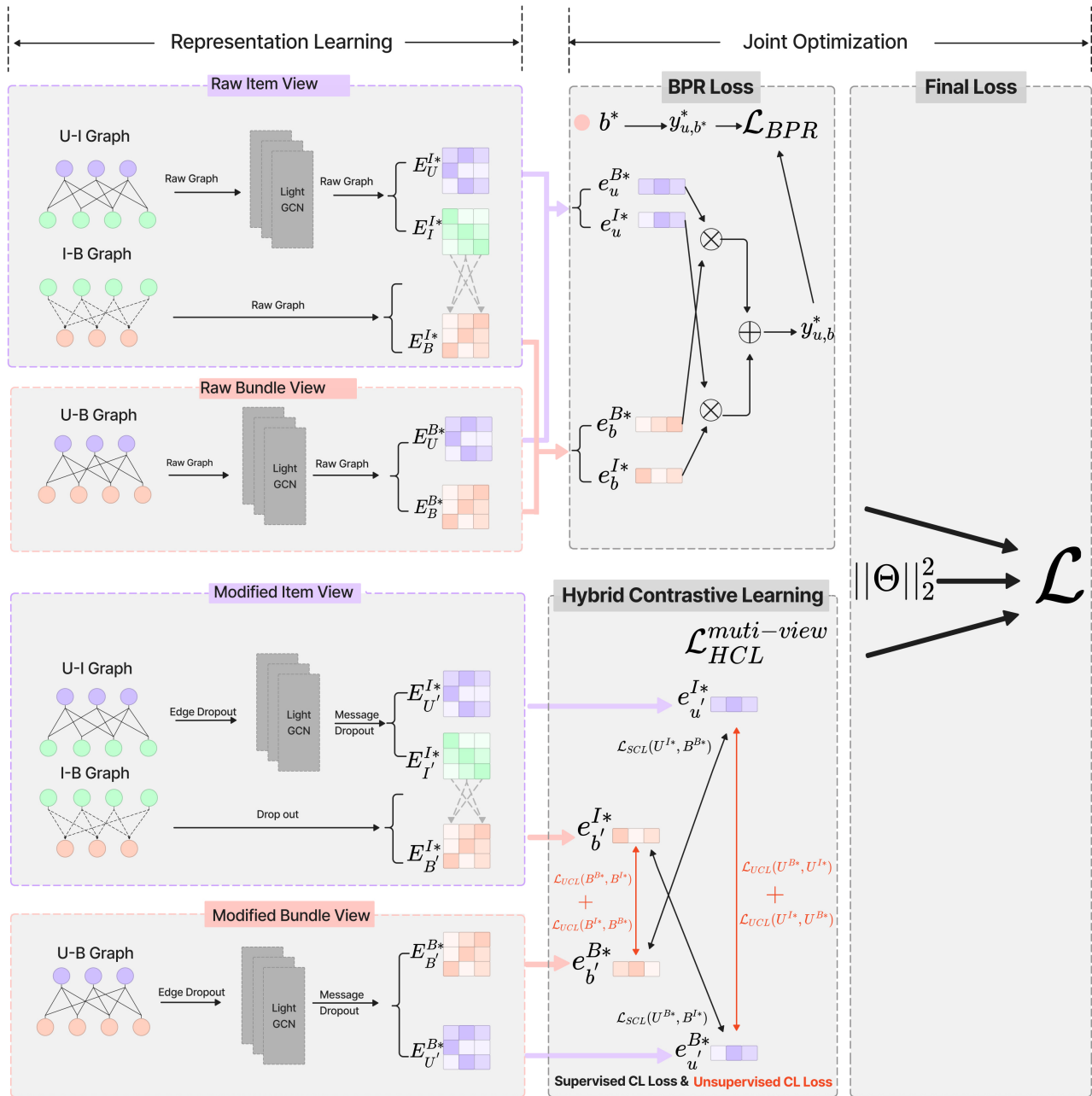


Figure 3. The overall framework of HCLBR consists of two main components: (1) Representation learning for both the user and bundle views, and (2) Joint optimization of the BPR loss \mathcal{L}_{BPR} and the hybrid contrastive loss \mathcal{L}_{HCL} .

the bundle representation $e_{b'}^{B*}$ generated from different views (see the numerator in Equation (9)). On the other hand, we aim to minimize the consistency between unobserved user-bundle pairs by uniformly sampling unobserved bundles for user u , thereby minimizing the consistency between user-bundle pairs (see the denominator in Equation (9)).

$$\mathcal{L}_{SCL}(U^{I*}, B^{B*}) = \sum_{(u', b') \in \mathcal{Y}} -\log \frac{\exp(f(e_{u'}^{I*}, e_{b'}^{B*})/t)}{\sum_{q' \in \mathcal{Q}} \exp(f(e_{u'}^{I*}, e_{q'}^{B*})/t)} \quad (9)$$

where, Q represents a set that includes an observed bundle for user u and a sampled set of unobserved bundles.

By doing so, the model is explicitly trained to learn the closeness of user-bundle interactions in diverse and imperfect views of users and bundles. This improves the model's robustness and generalizability. Please note that, in the context of supervised contrastive learning, we intuitively select to compute $\mathcal{L}_{SCL}(U^{B*}, B^{I*})$ on user and bundled package nodes from different views, instead of calculating $\mathcal{L}_{SCL}(U^{B*}, B^{I*})$ on user and bundle nodes from the same view. This is because calculating SCL loss on user and bundle nodes from the same view is redundant since they have already been utilized in computing the \mathcal{L}_{BPR} .

3.3.4. Multi-View Permutation

Recent work has demonstrated that computing the sum of contrastive tilt losses for all pairs of arbitrary two views across multiple incomplete views can improve overall performance [38]. Note that, in the denominator of the loss function $\mathcal{L}_{UCL}(U^{B*}, U^{I*})$, we fix a user node U^{B*} from the modified-item-view after data augmentation as an anchor and enumerate all user nodes U^{I*} in the modified-bundle-view after data augmentation. Hence, we also need to compute the symmetric $\mathcal{L}_{UCL}(U^{I*}, U^{B*})$ by anchoring on U^{I*} . Similarly, we calculate $\mathcal{L}_{SCL}(U^{B*}, B^{I*})$ and $\mathcal{L}_{SCL}(U^{I*}, B^{B*})$ for supervised contrastive learning. The overall multi-view HCL loss is the sum of unsupervised contrastive learning loss and supervised contrastive learning loss on user and bundle nodes.

$$\begin{aligned}\mathcal{L}_{HCL}^{multi-view} = & \mathcal{L}_{UCL}(U^{B*}, U^{I*}) + \mathcal{L}_{UCL}(U^{I*}, U^{B*}) \\ & + \mathcal{L}_{UCL}(B^{B*}, B^{I*}) + \mathcal{L}_{UCL}(B^{I*}, B^{B*}) \\ & + \mathcal{L}_{SCL}(U^{I*}, B^{B*}) + \mathcal{L}_{SCL}(U^{B*}, B^{I*}).\end{aligned}\quad (10)$$

We train the model using a multitask learning approach, where the final loss \mathcal{L} is a weighted combination of the supervised BPR ranking loss \mathcal{L}_{BPR} , the multi-view mixed contrastive learning loss $\mathcal{L}_{HCL}^{multi-view}$, and the $L2$ regularization term $\|\Theta\|_2^2$.

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{HCL}^{multi-view} + \lambda_2 \|\Theta\|_2^2 \quad (11)$$

where λ_1 and λ_2 are hyperparameters that balance the three components, and $\Theta = \{E_U^{B(0)}, E_B^{B(0)}, E_I^{I(0)}\}$ represents all the model parameters.

4. Experiments

4.1. Datasets

We evaluated the proposed architecture on four benchmark datasets: Youshu, NetEase, iFashion, and Steam. These datasets exhibit significant variations in size, bundled attributes (such as average bundle size), and domains, as shown in **Table 1**. Our findings demonstrate that HCLBR outperforms the state-of-the-art (SOTA) methods on diverse datasets, highlighting its significant advantage over existing solutions.

Table 1. Dataset statistics.

Dataset	#U	#I	#B	#U-I	#U-B	#Avg.I/B
YouShu	8039	32,770	4771	138,515	51,377	37.03
NetEase	18,528	123,628	22,864	1,128,065	302,303	77.80
iFashion	53,897	42,563	27,694	2,290,645	1,679,708	3.86
Steam	29,634	2819	615	902,967	87,568	5.77

YouShu is a Chinese book review website that allows users to create customized book lists [11]. In this dataset, bundles refer to sets of books generated by users.

NetEase is a cloud music service that enables consumers to choose songs and create playlists [21]. Intuitively, the bundles in this dataset correspond to playlists.

iFashion is an online fashion clothing recommendation dataset [39], where clothing items consisting of individual fashion pieces are considered as bundles. We followed the clothing recommendation setup [40] and preprocessed the iFashion dataset using user 20-core rules and clothing 10-core rules.

Steam is a video game website created by Valve, serving as a marketplace and distribution service. In this dataset, bundled games refer to different game bundles sold together on the website [41]. Interactions represent purchase events.

The statistical data of the datasets is outlined in Table 1. It is observed that the item order within bundles is either irrelevant or not provided across all datasets. Additionally, auxiliary information is exclusively available in the Steam dataset. However, we intentionally disregarded it to maintain a consistent collaborative filtering (CF) approach across all datasets. We acknowledge that traditional content-based filtering methods [42] can be readily employed to incorporate the support of auxiliary information if desired.

4.2. Experimental Setup

4.2.1. Evaluation Metrics

In practical applications, TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) are used to define the sets in a confusion matrix. The positive class is typically assigned to important and rare indicators, while the negative class is assigned to less important and common indicators. This is because the detection of defective items and the identification of fraudulent instances are of particular interest. The metrics of interest include the proportion of detected defective items and the proportion of falsely classified non-defective items. These metrics are crucial for assessing the model's performance and are often referred to as "Recall". Recall represents the ratio of detected defective/fraudulent instances, providing valuable insights into the model's effectiveness.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (12)$$

Regarding Recall@K, it refers to the ratio of the number of relevant results retrieved in the top K results to the total number of relevant results in the database. It measures the retrieval system's ability to retrieve all relevant results. Formally, it can be defined as follows:

$$\text{Recall@K} = \frac{\text{TP@K}}{\text{TP@K} + \text{FN@K}} \quad (13)$$

Normalized Discounted Cumulative Gain (NDCG) is an evaluation metric that takes into account the ranking order of the returned results. It is a normalized measure with values ranging from 0 to 1, where higher values indicate better performance.

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} \quad (14)$$

where DCG (Discounted Cumulative Gain) is a measure of cumulative gain that incorporates a discounting factor.

$$\text{DCG@K} = \sum_{i=1}^K \frac{rel_i}{\log_2(i+1)} \quad (15)$$

where rel_i refers to the true relevance score of the i -th result.

$$\text{IDCG@K} = \sum_{i=1}^{|REL|} \frac{rel_i}{\log_2(i+1)} \quad (16)$$

where IDCG (Ideal DCG) represents the ideal DCG value. $|REL|$ denotes the number of results in the set formed by taking the top K results when sorting based on true relevance scores in descending order.

For each user, we classify projects that the user interacts with as positive projects, and projects that the user does not interact with as negative projects. Recall@K and NDCG@K are utilized as evaluation metrics, with K values ranging from 5, 10, 20, to 40. NDCG@20 is employed for selecting the optimal model based on the validation set and ranking all projects during testing [18].

4.2.2. Baselines

We compared HCLBR with four benchmark models, including the state-of-the-art (SOTA) approach. For each of these models, we used publicly available repositories as described below:

BPR [12]: It is a Bayesian Personalized Ranking method that employs matrix factorization in a pairwise learning framework. This algorithm ranks items based on implicit feedback from user interactions with bundles.

DAM [11]: This is a powerful baseline that utilizes attention mechanisms to learn bundle representations and incorporates multitask learning to optimize user-item and user-bundle interactions. The authors have demonstrated its superiority over BPR, Neural Collaborative Filtering [18], and Embedding Factorization Machines [21] on platforms such as NetEase and YouShu.

BGCN [13]: BGCN decomposes the user-bundle-item relationships into two separate views by constructing bundle-view and item-view graphs. It leverages Graph Convolutional Networks (GCN) to learn representations and predict the relationships between users and bundles.

BRUCE [22]: It employs Transformers technology to simulate users' bundle preferences and the relationships between items that form a bundle, resulting in an improved performance compared to the benchmarks.

CrossCBR [15]: The current SOTA model leverages cross-view contrastive learning to simulate cross-view collaborative associations in bundle recommendation. It introduces cross-view contrastive learning to effectively regulate cross-view representations, resulting in substantial improvements over the benchmark models.

We also attempted to include BundleNet [14] as a baseline. However, since the code for BundleNet is not available online, we were unable to reproduce their results.

4.2.3. Parameter Settings

For all methods, the embedding size is set to 64, and Xavier normal initialization [43] is applied. The implementation is done using PyTorch [44] and the Adam optimizer [45]. BRUCE is trained with a learning rate of 0.001 (without warm-up) and a batch size of 2048. Regarding our method, we tune the hyperparameters $K, \lambda_1, \lambda_2, \tau$, and ρ within the following ranges: $\{1, 2, 3\}$, $\{0.01, 0.04, 0.1, 0.5, 1\}$, $\{10^{-6}, 10^{-5}, 2 \times 10^{-5}, 4 \times 10^{-5}, 10^{-4}\}$, $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5\}$, and $\{0, 0.1, 0.2, 0.5\}$. For the baseline methods, we refer to the reported results in [22] for BPR, DAM, BGCN, and BRUCE on the YouShu, NetEase, and Steam datasets, as their settings align with ours. We independently implemented all other baseline models and experimented with learning rates ranging from 0.0001 to 0.1, as well as embedding sizes ranging from 16 to 64.

All models were trained using PyTorch 1.11.0, with 64GB of memory, on an NVIDIA RTX3080 GPU.

4.3. Comparison with the Baseline Methods

We first compare the overall recommendation performance of HCLBR with user-item recommendation baselines and bundle-specific recommendation baselines on four datasets, as presented in **Table 2**. The best-performing method is highlighted in bold, while the strongest baseline is indicated with an underline. The column “%improv.” represents the relative improvement of HCLBR compared to the strongest baseline. From the results, we make the following observations.

- HCLBR consistently outperforms all baselines across datasets and metrics, demonstrating its superiority in bundle recommendation. By incorporating contrastive learning, HCLBR captures rich item representations and relationships, enabling a better understanding of bundle characteristics and more

Table 2. Model performance comparison on public datasets.

Method	NetEase						Youshu					
	Recall @5	NDCG @5	Recall @10	NDCG @10	Recall @20	NDCG @20	Recall @5	NDCG @5	Recall @10	NDCG @10	Recall @20	NDCG @20
BPR	0.0131	0.0107	0.0212	0.0139	0.0363	0.0188	0.0764	0.0638	0.1034	0.0729	0.1532	0.0890
DAM	0.0099	0.0077	0.0163	0.0102	0.0293	0.0142	0.1098	0.0873	0.1460	0.0996	0.2000	0.1162
BGCN	0.0138	0.0115	0.0244	0.0154	0.0427	0.0210	0.0942	0.0880	0.1401	0.1002	0.1992	0.1170
BRUCE	0.0183	0.0145	0.0296	0.0189	0.0489	0.0251	<u>0.1344</u>	0.1068	<u>0.1893</u>	0.1249	0.2624	0.1471
CrossCBR	<u>0.0322</u>	<u>0.0285</u>	<u>0.0521</u>	<u>0.0357</u>	<u>0.0824</u>	<u>0.0451</u>	0.1231	<u>0.1336</u>	0.1889	<u>0.1384</u>	<u>0.2753</u>	<u>0.1637</u>
HCLBR	0.0343	0.0296	0.0555	0.0369	0.0852	0.0462	0.1361	0.1265	0.1982	0.1441	0.2845	0.1680
Improv. (%)	6.12	3.72	6.13	3.25	3.29	2.38	1.25	10.20	4.49	3.96	3.23	2.56

Method	Steam						iFashion					
	Recall @5	NDCG @5	Recall @10	NDCG @10	Recall @20	NDCG @20	Recall @5	NDCG @5	Recall @10	NDCG @10	Recall @20	NDCG @20
BPR	0.4254	0.3693	0.4523	0.3788	0.4861	0.3882	0.0181	0.0186	0.0265	0.0229	0.0434	0.0300
DAM	0.6561	0.4740	0.9216	0.5691	0.9660	0.5817	0.0248	0.0210	0.0397	0.0271	0.0629	0.0450
BGCN	0.6078	0.4886	0.8062	0.5576	0.9374	0.5935	0.0254	0.0219	0.0427	0.0286	0.0693	0.0474
BRUCE	0.8799	0.7343	0.9568	0.7622	0.9806	0.7701	0.0423	<u>0.0596</u>	<u>0.0732</u>	<u>0.0696</u>	<u>0.1094</u>	0.0776
CrossCBR	<u>0.0903</u>	<u>0.7963</u>	<u>0.9657</u>	<u>0.8175</u>	<u>0.9851</u>	<u>0.8139</u>	<u>0.0451</u>	0.0576	0.0712	0.0672	0.1078	<u>0.0829</u>
HCLBR	0.9560	0.8481	0.9807	0.8578	0.9906	0.8600	0.0492	0.0625	0.0756	0.0715	0.1135	0.0864
Improv. (%)	4.88	6.11	1.53	4.70	0.56	5.36	8.33	7.84	5.82	6.01	3.61	4.05

accurate recommendations. It's important to note that CrossCBR, on the other hand, has limitations as it only generates views with edge losses, which restricts downstream contrastive learning. This emphasizes the significance of incorporating contrastive learning techniques and utilizing multiple views for improved recommendation accuracy. Overall, HCLBR's comprehensive approach leverages contrastive learning in both unsupervised and supervised learning environments, allowing it to capture nuanced relationships and significantly enhance bundle recommendations.

- BGCN, in the realm of general user project recommendation models, consistently outperforms BPR, indicating that GNN-based methods are effective in capturing user bundle CF signals and enhancing recommendations.
- BRUCE consistently outperforms BGCN in all datasets and metrics, highlighting the effectiveness of Transformer-based models in improving the accuracy of recommendation systems.

- CrossCBR consistently outperforms BRUCE across a wide range of datasets and metrics, clearly demonstrating the effectiveness of graph contrastive learning in recommendation tasks. This strongly emphasizes the valuable supplementary information offered by item views and their potential to enhance the model's ability to discriminate and make accurate recommendations.

4.4. Ablation Study

4.4.1. Effectiveness of Data Augmentations

During the representation learning process for two views, we conducted experiments with various data augmentation settings. HCLBR_RW represents the original preservation (*i.e.*, no augmentation), HCLBR_ED stands for Edge Dropout, an augmentation method based on graphs, and HCLBR_MD represents Message Dropout, an augmentation method based on embeddings. The results in Figure 4 indicate that the differences between the three data augmentation settings of HCLBR can be considered negligible compared to the performance gain of the baseline. This finding suggests that the inherent differences within the original data of multiple views provide sufficient variance for mixed multi-view contrastive learning, while the variance introduced by random data augmentation is inconsequential. In the future, further exploration can be done to discover more advanced and effective data augmentation methods.

4.4.2. Effect of the Number of Embedding Sizes

When evaluating the performance of a recommendation system at different embedding sizes, we observed interesting results. In the Youshu dataset, we did not observe a significant change in NDCG@20 and Recall@20 values as the embedding size increased. The values fluctuated slightly with increasing embedding dimension, but the overall trend was not clear. This suggests that increasing the number of embedding dimensions did not significantly improve the performance of the recommendation system for this dataset. This observation is also reflected in Figure 5.

However, in the NetEase dataset, we found that NDCG@20 and Recall@20 values increased as the embedding size increased. This indicates that increasing the number of embedding dimensions can significantly improve the performance of the recommendation system for this dataset.

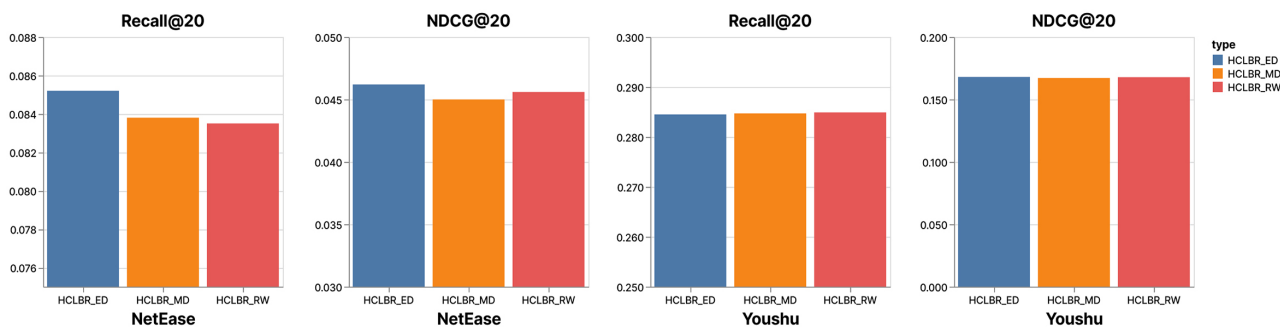


Figure 4. A comparison of the results obtained from various data augmentation methods on NetEase and Youshu platforms.

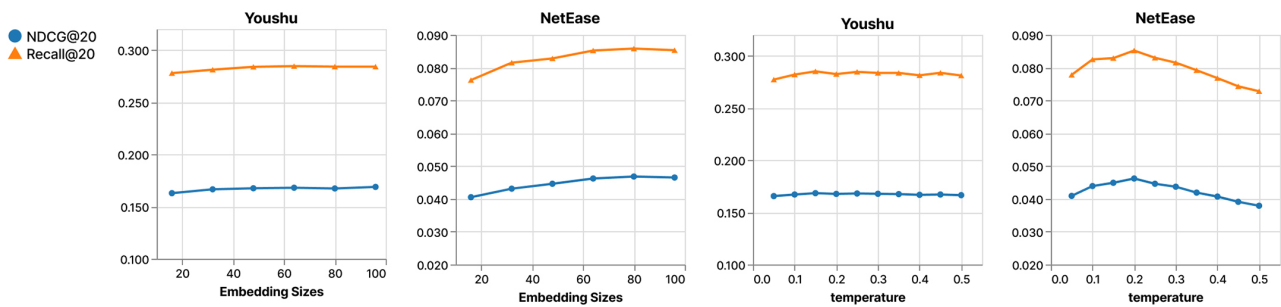


Figure 5. HCLBR's NDCG@20 and Recall@20 performance varies with temperature and embedding sizes on both the NetEase and Youshu datasets.

Overall, our ablation study suggests that increasing the number of embedding dimensions may have an impact on the performance of a recommendation system, but this effect depends on the specific dataset. Therefore, when designing and optimizing recommendation systems, we need to consider the properties of the dataset and conduct appropriate experiments to determine the optimal number of embedding dimensions.

4.5. Hyper-Parameter Tuning and Computational Efficiency Analysis

4.5.1. Hyper-Parameter Study

Our experiments demonstrate that the temperature hyperparameter has a significant impact on the performance of HCLBR, as indicated by NDCG@20 and Recall@20. To investigate this effect, we set the value of τ to range from 0.05 to 0.5 and studied the performance of our HCLBR model across this range of values. Specifically, our results on the NetEase dataset show that HCLBR is sensitive to deviations from the optimal temperature setting, and such deviations can lead to a decrease in performance. Conversely, the effect of temperature on HCLBR's performance on the Youshu dataset is less pronounced. Therefore, when tuning the temperature hyperparameter for HCLBR, it is crucial to consider the characteristics of the specific dataset and conduct appropriate experiments to determine the optimal setting.

4.5.2. Computational Efficiency

In order to evaluate the computational efficiency of our model, we compared it and two of its variants with BGCN and BRUCE in terms of training time over a period of 10 consecutive epochs. We recorded the training time for each epoch and calculated the average training time per epoch, as shown in **Table 3**. Our analysis revealed two important findings. Firstly, the efficiency of our model, HCLBR, was significantly higher than that of BGCN, demonstrating its superior computational efficiency. However, BRUCE had a faster training speed due to its use of a pre-trained model, which can take up a considerable amount of time during pre-training. And our model exhibits a modest enhancement in computational efficiency when compared to CrossCBR. Secondly, we compared two variants of our model, HCLBR_ED and HCLBR_RW, and found that data

Table 3. The statistics of one-epoch training time (seconds) for HCLBR and baselines on RTX3080, where the “HCL” is short of “HCLBR”.

RTX3080	BGCN	CrossCBR	BRUCE	HCL_ED	HCL_RW	HCL_16
Youshu	23.76	3.15	1.22	2.90	3.20	3.10
NetEase	80.59	15.23	10.23	14.78	19.27	13.42

augmentation during training was consistent with our hypothesis. We also found that node dropout could optimize training time to some extent, particularly on larger networks. When comparing HCLBR_16 with the other models, we found that the difference in training time was not significant, indicating that the embedding size had little impact on training time.

5. Conclusion and Future Work

In this study, we have proposed HCLBR, a novel bundle recommendation method that leverages multi-view hybrid contrastive learning. Our approach effectively captures collaborative associations and enhances perspective-aware representations, leading to improved recommendation performance. HCLBR considers bundle and item views as separate yet intertwined perspectives of user-bundle preference. By applying contrastive learning to these views, HCLBR transforms their coherence into representations of users and bundles. Through comprehensive evaluations on four benchmark datasets, we have demonstrated that HCLBR significantly outperforms state-of-the-art methods. This highlights the effectiveness of our proposed method in enhancing bundle recommendation accuracy. Furthermore, our ablation and model analyses have provided valuable insights into the underlying mechanisms driving the substantial performance improvement achieved by HCLBR. These insights can guide future research in the field of bundle recommendation and emphasize the importance of considering multiple perspectives and leveraging contrastive learning techniques. Looking ahead, our future work will explore the use of transformers for bundle generation tasks and emphasize the need for explainability and interpretability in recommendation systems. By delving deeper into these aspects, we aim to further enhance the performance and practicality of bundle recommendation systems.

Acknowledgments

This research was supported in part by the Scientific Research Foundation of Sichuan University of Science and Engineering (2021RC13) and in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars of the Department of Human Resources and Social Security of Sichuan Province.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Bakos, Y. and Brynjolfsson, E. (1999) Bundling Information Goods: Pricing, Profits, and Efficiency. *Management Science*, **45**, 1613-1630. <https://doi.org/10.1287/mnsc.45.12.1613>
- [2] Gaeth, G.J., Levin, I.P., Chakraborty, G. and Levin, A.M. (1991) Consumer Evaluation of Multi-Product Bundles: An Information Integration Analysis. *Marketing Letters*, **2**, 47-57. <https://doi.org/10.1007/BF00435195>
- [3] Hanson, W. and Martin, R.K. (1990) Optimal Bundle Pricing. *Management Science*, **36**, 123-246. <https://doi.org/10.1287/mnsc.36.2.155>
- [4] Stremersch, S. and Tellis, G.J. (2002) Strategic Bundling of Products and Prices: A New Synthesis for Marketing. *Journal of Marketing*, **66**, 55-72. <https://doi.org/10.1509/jmkg.66.1.55.18455>
- [5] Garfinkel, R., Gopal, R., Tripathi, A. and Yin, F. (2006) Design of a Shopbot and Recommender System for Bundle Purchases. *Decision Support Systems*, **42**, 1974-1986. <https://doi.org/10.1016/j.dss.2006.05.005>
- [6] Ge, Y., Xiong, H., Tuzhilin, A. and Liu, Q. (2014) Cost-Aware Collaborative Filtering for Travel Tour Recommendations. *ACM Transactions on Information Systems*, **32**, 1-31. <https://doi.org/10.1145/2559169>
- [7] Xie, M., Lakshmanan, L.V.S. and Wood, P.T. (2010) Breaking out of the Box of Recommendations. *Proceedings of the Fourth ACM Conference on Recommender Systems*, Barcelona, 26-30 September 2010, 151-158. <https://doi.org/10.1145/1864708.1864739>
- [8] Bai, J., Zhou, C., Song, J., Qu, X., An, W., Li, Z. and Gao, J. (2019) Personalized Bundle List Recommendation. *The World Wide Web Conference*, San Francisco, 13-17 May 2019, 60-71. <https://doi.org/10.1145/3308558.3313568>
- [9] He, Y., Zhang, Y., Liu, W. and Caverlee, J. (2020) Consistency-Aware Recommendation for User-Generated Item List Continuation. *Proceedings of the 13th International Conference on Web Search and Data Mining*, Houston, 3-7 February 2020, 250-258. <https://doi.org/10.1145/3336191.3371776>
- [10] Liu, Y., Xie, M. and Lakshmanan, L.V.S. (2014) Recommending User Generated Item Lists. *Proceedings of the 8th ACM Conference on Recommender Systems*, Foster City, 6-10 October 2014, 185-192. <https://doi.org/10.1145/2645710.2645750>
- [11] Chen, L., Liu, Y., He, X., Gao, L. and Zheng, Z. (2019) Matching User with Item Set: Collaborative Bundle Recommendation with Deep Attention Network. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, 10-16 August 2019, 2095-2101. <https://doi.org/10.24963/ijcai.2019/290>
- [12] Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L. (2012) BPR: Bayesian Personalized Ranking from Implicit Feedback. arXiv: 1205.2618.
- [13] Chang, J., Gao, C., He, X., Jin, D. and Li, Y. (2020) Bundle Recommendation with Graph Convolutional Networks. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 25-30 July 2020, 1673-1676. <https://doi.org/10.1145/3397271.3401198>
- [14] Deng, Q., Wang, K., Zhao, M., Zou, Z., Wu, R., Tao, J., Fan, C. and Chen, L. (2020) Personalized Bundle Recommendation in Online Games. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 19-23 October 2020, 2391-2399. <https://doi.org/10.1145/3340531.3412734>
- [15] Ma, Y., He, Y., Zhang, A., Wang, X. and Chua, T.S. (2022) CrossCBR: Cross-View Contrastive Learning for Bundle Recommendation. *Proceedings of the 28th ACM*

- SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington DC, 14-18 August 2022, 1233-1241. <https://doi.org/10.1145/3534678.3539229>
- [16] Chen, W., Gu, Y., Ren, Z., He, X., Xie, H., Guo, T., Yin, D. and Zhang, Y. (2019) Semi-Supervised User Profiling with Heterogeneous Graph Attention Networks. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, 10-16 August 2019, 2116-2122. <https://doi.org/10.24963/ijcai.2019/293>
- [17] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y. and Wang, M. (2020) LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 25-30 July 2020, 639-648. <https://doi.org/10.1145/3397271.3401063>
- [18] Wang, X., He, X., Wang, M., Feng, F. and Chua, T.S. (2019) Neural Graph Collaborative Filtering. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Paris, 21-25 July 2019, 165-174. <https://doi.org/10.1145/3331184.3331267>
- [19] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L. and Leskovec, J. (2018) Graph Convolutional Neural Networks for Web-Scale Recommender Systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, 19-23 August 2018, 974-983. <https://doi.org/10.1145/3219819.3219890>
- [20] Ding, Y., Ma, Y., Wong, W.K. and Chua, T.S. (2021) Leveraging Two Types of Global Graph for Sequential Fashion Recommendation. *Proceedings of the 2021 International Conference on Multimedia Retrieval*, Taipei, 21-24 August 2021, 73-81. <https://doi.org/10.1145/3460426.3463638>
- [21] Cao, D., Nie, L., He, X., Wei, X., Zhu, S. and Chua, T.S. (2017) Embedding Factorization Models for Jointly Recommending Items and User Generated Lists. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tokyo, 7-11 August 2017, 585-594. <https://doi.org/10.1145/3077136.3080779>
- [22] Avny Brosh, T., Livne, A., Sar Shalom, O., Shapira, B. and Last, M. (2022) BRUCE: Bundle Recommendation Using Contextualized Item Embeddings. *Sixteenth ACM Conference on Recommender Systems*, Seattle, 18-23 September 2022, 237-245. <https://doi.org/10.1145/3523227.3546754>
- [23] Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J. and Xie, X. (2021) Self-Supervised Graph Learning for Recommendation. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 11-15 July 2021, 726-735. <https://doi.org/10.1145/3404835.3462862>
- [24] Gomez, H., Jusinkas, R.L. and Lipstein, A. (2021) Cosmological Scattering Equations. *Physical Review Letters*, **127**, Article ID: 251604. <https://doi.org/10.1103/PhysRevLett.127.251604>
- [25] Lofnes, I.M. (2022) Quarkonia as Probes of the QGP and of the Initial Stages of the Heavy-Ion Collision with ALICE. *EPJ Web of Conferences*, **259**, Article No. 12004. <https://doi.org/10.1051/epjconf/202225912004>
- [26] Klimashevskaja, A., Elahi, M. and Trattner, C. (2023) Addressing Popularity Bias in Recommender Systems: An Exploration of Self-Supervised Learning Models. *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, Limassol, 26-29 June 2023, 7-11. <https://doi.org/10.1145/3563359.3597409>

- [27] Yang, Y., Huang, C., Xia, L. and Li, C. (2022) Knowledge Graph Contrastive Learning for Recommendation. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Madrid, 11-15 July 2022, 1434-1443. <https://doi.org/10.1145/3477495.3532009>
- [28] Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J. and Yin, D. (2019) Graph Neural Networks for Social Recommendation. *The World Wide Web Conference*, San Francisco, 13-17 May 2019, 417-426. <https://doi.org/10.1145/3308558.3313488>
- [29] Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Zhang, J., Ding, B. and Cui, B. (2022) Contrastive Learning for Sequential Recommendation. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, Kuala Lumpur, 9-12 May 2022, 1259-1273. <https://doi.org/10.1109/ICDE53745.2022.00099>
- [30] Qiu, R., Huang, Z., Yin, H. and Wang, Z. (2022) Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 21-25 February 2022, 813-823. <https://doi.org/10.1145/3488560.3498433>
- [31] Xia, X., Yin, H., Yu, J., Shao, Y. and Cui, L. (2021) Self-Supervised Graph Co-Training for Session-Based Recommendation. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1-5 November 2021, 2180-2190. <https://doi.org/10.1145/3459637.3482388>
- [32] Chen, Y., Liu, Z., Li, J., McAuley, J. and Xiong, C. (2022) Intent Contrastive Learning for Sequential Recommendation. *Proceedings of the ACM Web Conference 2022*, Lyon, 25-29 April 2022, 2172-2182. <https://doi.org/10.1145/3485447.3512090>
- [33] Wang, T. and Isola, P. (2020) Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. *International Conference on Machine Learning*, 12-18 July 2020, 9929-9939.
- [34] Chen, T., Kornblith, S., Norouzi, M. and Hinton, G. (2020) A Simple Framework for Contrastive Learning of Visual Representations. *International Conference on Machine Learning*, 12-18 July 2020, 1597-1607.
- [35] Gao, T., Yao, X. and Chen, D. (2021) SimCSE: Simple Contrastive Learning of Sentence Embeddings. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Punta Cana, November 2021, 6894-6910. <https://doi.org/10.18653/v1/2021.emnlp-main.552>
- [36] Liu, Z., Ma, Y., Ouyang, Y. and Xiong, Z. (2021) Contrastive Learning for Recommender System. arXiv: 2101.01317.
- [37] Gutmann, M. and Hyvärinen, A. (2010) Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Sardinia, 13-15 May 2010, 297-304.
- [38] Tian, Y., Krishnan, D. and Isola, P. (2020) Contrastive Multiview Coding. In: Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.M., Eds., *ECCV 2020: Computer Vision—ECCV 2020*, Springer, Cham, 776-794. https://doi.org/10.1007/978-3-030-58621-8_45
- [39] Chen, W., Huang, P., Xu, J., Guo, X., Guo, C., Sun, F., Li, C., Pfadler, A., Zhao, H. and Zhao, B. (2019) POG: Personalized Outfit Generation for Fashion Recommendation at Alibaba iFashion. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, 4-8 August 2019, 2662-2670. <https://doi.org/10.1145/3292500.3330652>
- [40] Li, X., Wang, X., He, X., Chen, L., Xiao, J. and Chua, T.S. (2020) Hierarchical Fashion Graph Network for Personalized Outfit Recommendation. *Proceedings of the*

43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 25-30 July 2020, 159-168.

<https://doi.org/10.1145/3397271.3401080>

- [41] Pathak, A., Gupta, K. and McAuley, J. (2017) Generating and Personalizing Bundle Recommendations on Steam. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tokyo, 7-11 August 2017, 1073-1076. <https://doi.org/10.1145/3077136.3080724>
- [42] Melville, P., Mooney, R.J. and Nagarajan, R. (2002) Content-Boosted Collaborative Filtering for Improved Recommendations. *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, Edmonton, July 2002, 187-192.
- [43] Glorot, X. and Bengio, Y. (2010) Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Sardinia, 13-15 May 2010, 249-256.
- [44] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N. and Antiga, L. (2019) Pytorch: An Imperative Style, High-Performance Deep Learning Library. *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, 8-14 December 2019, 8024-8035.
- [45] Kingma, D.P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. arXiv: 1412.6980.