Data-Augmented Counterfactual Learning for Bundle Recommendation

Shixuan Zhu  
Tongji University  
Shanghai, China  
2130768@tongji.edu.cn

Qi Shen  
Tongji University  
Shanghai, China  
1653282@tongji.edu.cn

Yiming Zhang  
Tongji University  
Shanghai, China  
2030796@tongji.edu.cn

Zhenwei Dong  
Tongji University  
Shanghai, China  
1853155@tongji.edu.cn

Zhihua Wei*  
Tongji University  
Shanghai, China  
zhihua_weii@tongji.edu.cn

ABSTRACT

Bundle Recommendation (BR) aims at recommending bundled items on online content or e-commerce platform, such as song lists on a music platform or book lists on a reading website. Several graph-based models have achieved state-of-the-art performance on BR task. But their performance is still sub-optimal, since the data sparsity problem tends to be more severe in real bundle recommendation scenarios, which limits graph-based models from more sufficient learning. In this paper, we propose a novel graph learning paradigm called Counterfactual Learning for Bundle Recommendation (CLBR) to mitigate the impact of data sparsity problem and improve bundle recommendation. Our paradigm consists of two main parts: counterfactual data augmentation and counterfactual constraint. The main idea of our paradigm lies in answering the counterfactual questions: “What would a user interact with if his/her interaction history changes?” “What would a user interact with if the bundle-item affiliation relations change?” In counterfactual data augmentation, we design a heuristic sampler to generate counterfactual graph views for graph-based models, which has better noise controlling than the stochastic sampler. We further propose counterfactual loss to constrain model learning for mitigating the effects of residual noise in augmented data and achieving more sufficient model optimization. Further theoretical analysis demonstrates the rationality of our design. Extensive experiments of BR models applied with our paradigm on two real-world datasets are conducted to verify the effectiveness of the paradigm.

CCS CONCEPTS

• Information systems → Recommender systems.

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06. $15.00
https://doi.org/XXXXXXX.XXXXXX

1 INTRODUCTION

Bundled items are very common on nowadays online content or e-commerce platforms, which makes Bundle Recommendation (BR) [33] become an important task. The task aims to recommend a bundle with multiple items for users, which can improve users’ experience for providing more variable options, as well as increase business profits for the expansion of order size. Typical BR scenarios include book lists recommendation on online reading websites [13], song lists recommendation on music streaming platforms, video collections recommendation on online video websites, etc.

On the basis of general recommendation methods, most existing works for BR introduce user-item interaction and bundle-item affiliation information as supplements for user-bundle interaction. For utilizing these key information, graph-based models are employed[2, 17] to encode them into user and bundle embeddings [5, 8, 14], and some multi-task frameworks [3, 6, 8] are leveraged for model training.

However, the effectiveness of the aforementioned graph-based BR models usually depends on enough high-quality training data. Unfortunately, the data sparsity problem is more severe and complex in most realistic BR scenarios than in general recommendation, since there exists not only user-side sparsity but also creator-side sparsity problems. For the users of platform, similar to the ubiquitous user-item interaction sparsity [10, 27, 30], the user-bundle interactions are also very sparse. For the bundle creators, although there are massive possible item combinations for bundle generation, very few bundles are actually created and shared [7], which leads to sparse bundle-item affiliations. These prevent adequate training for the graph-based BR models, and lead to sub-optimal recommendation results.

For confronting these data sparsity in BR, we focus on the counterfactual thinking in causal theory and resort to counterfactual data as a complement to the observed sparse data. In counterfactual thinking, the real data we observe in BR scenarios, whether
interaction data or bundle-item affiliation relation, is only a small fraction of all possible data, or counterfactual data. For instance, Figure 1 illustrates a toy example of BR in E-commerce, where a user can explore not only products, but also shopping carts that consist of multiple products created by other users. We observe that a user purchased a keyboard instead of a mouse, and only chose bundle \( A \) and bundle \( B \). But in the counterfactual world, due to possible different exposure mechanisms, the user may have bought the mouse, and additionally selected bundle \( C \) that has similar contents with bundle \( A \). Or perhaps the contents of bundle \( B \) changes due to the substitutability of some items, but the user still can accept the changed bundle. These counterfactual data can more comprehensively reveal users’ preference, which naturally solve the data sparsity problem and contribute to more effective model training.

Motivated by the analysis above, we start with the following “what if” questions: “What would a user interact with if his/her interaction history changes?” “What would a user interact with if the bundle-item affiliation relations change?” and try to simulate these counterfactuals by intervening on the user’s interaction history and bundle-item affiliation for improving BR model training. We propose a novel learning paradigm called Counterfactual Learning for Bundle Recommendation (CLBR) to generate counterfactual data and guide the graph-based BR model to utilize these data effectively. It consists of two main parts: counterfactual data augmentation and counterfactual constraint. For the data augmentation, we develop a simple yet efficient heuristic sampler to generate counterfactual graph views based on the original BR graph, which achieves effective noise control compared to the stochastic sampler. The counterfactual constraint helps models discard intractable noise in the generated data that deviates from the counterfactual distribution, and make efficient use of augmented data to learn accurate and robust representations for BR. We apply the CLBR paradigm to several state-of-the-art graph-based models. Extensive experiments on two real-world datasets demonstrate the effectiveness of the proposed paradigm.

Our main contributions in this work are summarized below:

- We introduce the data sparsity problem in bundle recommendation, and try to leverage the counterfactual thinking into bundle recommendation for confronting the problem.
- We propose Counterfactual Learning for Bundle Recommendation (CLBR), a novel learning paradigm for graph-based models, which includes counterfactual data augmentation with a heuristic sampler and counterfactual constraint for training, to achieve more sufficient model optimization.
- We theoretically analyze the noisy information in the augmented data, and interpret our proposed constraint term via information theory, to verify the rationality of our design.
- We apply CLBR on several state-of-the-art graph-based models and conduct extensive experiments based on real-world BR datasets, which verify the effectiveness of the paradigm.

2 RELATED WORKS

Bundle Recommendation. Recently, several efforts have been made in Bundle Recommendation, which recommends a bundle of items that users may have interest to interact together\([4, 6, 11, 12, 14, 16]\). Sar Shalom et al. \([16]\) introduced the list recommendation problem, and optimized a list’s click probability based on collaborative filtering. Liu et al. \([12]\) estimated the probability that a consumer would buy an item associated with already bought item to recommend product bundles. EFM \([4]\) captured users’ preferences over items and lists by factorization model. DAM \([6]\) contributed a factorized attention network to gather item information into bundle representations in a multi-task way jointly for user-bundle and user-item interactions. More recently, graph-based recommendation models have become the state-of-the-art methods for BR. HPGN \([11]\) proposed a Hierarchical Fashion Graph Neural Network to obtain more expressive representations for users and outfits, which incorporated both item-level and bundle-level semantics into the bundle representations. BGCN \([5]\) utilized graph neural network (GNN) to learn representations from user-level and bundle-level graph views, and CrossCBR\([14]\) further adopted contrastive learning to model the cooperative association between these two views to achieve mutual enhancement.

However, previous researches rely on large amount of reliable data. The severe data sparsity problems existed in BR scenarios limit the performance of graph-based models, which inspires the study of counterfactual data augmentation in this paper.

Counterfactual View in Recommendation. As a key theory of causal inference, counterfactual paradigm focuses on estimating causal effects with \( \text{What if} \) problems and has been widely adopted in many machine learning domains recently, including computer vision \([1, 9]\), natural language processing \([34]\) and graph mining \([19, 32]\). In the recommendation field, some works leveraged counterfactual thinking for debias problem. For example, Wang et al. \([22]\) estimated the counterfactual click likelihood of the user for reducing the direct effect of exposure features and eliminating the clickbait issue. Zhang et al. \([31]\) deconfounded the popularity bias in training phase, and adjusted the recommendation score with desired popularity information via causal intervention during inference phase. Besides, some researchers improved the recommendation models in causal view. Zhang et al. \([29]\) empowered attention mechanism by the causal regularization, which uses individual treatment effect to measure the causal relation. In addition, some methods explored the counterfactual in data-augmentation manner \([26]\). For instance, Wang et al. \([23]\) proposed data- and model-oriented sampling methods for generating better counterfactual sequences when training the anchor model. Xiong et al. \([24]\) designed a learning-based method to discover more effective

Figure 1: A toy example of bundle recommendation in E-commerce, where the bundles are shopping carts that consist of multiple products.
samples, actively intervened on the user preference, and predicted the user feedback based on a pre-trained recommender.

However, the counterfactual view in the heterogeneous graph field, e.g. the complex multi-relation in BR, is of great need but largely unexplored.

3 PRELIMINARY

3.1 Task Definition

Given user set $U = \{u_1, u_2, \ldots, u_N\}$, item set $V = \{v_1, v_2, \ldots, v_L\}$, and bundle set $B = \{b_1, b_2, \ldots, b_K\}$, we formulate the user-item interaction data as $E_{uv} = \{(u, v) | u \text{ interacted } v, u \in U, v \in V\}$, user-bundle interaction data as $E_{ub} = \{(u, b) | u \text{ interacted } b, u \in U, b \in B\}$ and bundle-item affiliation relation as $E_{bv} = \{(b, v) | v \text{ belongs to } b, b \in B, v \in V\}$ . Based on the above information, bundle recommender estimates the probability that user will interact with each candidate bundle.

3.2 A Unified Framework

Then we summarize previous bundle recommendation methods into a unified graph neural network framework for subsequent leveraging. Generally, we organize the user, item and bundle and their relation information as a heterogeneous user-bundle-item graph $G = (\mathcal{V}, \mathcal{E})$ following [5, 11], with initialized node embeddings $X \in \mathbb{R}^{(N+L+K) \times D}$ of $\mathcal{V}$, where $D$ is the embedding dimension. Then we use the unified user-bundle-item relation graph defined above as the basis of graph-based BR models. The general structure of existing graph-based BR models can be summarized in the following two parts: (1) modeling the cross-semantic relation of user, item and bundle; (2) calculating the final matching score of user and bundle. Therefore, we formula the unified bundle recommender $M_B(\mathbf{X}, \mathbf{G})$ mainly as a graph-based encoder and an embedding-based predictor:

$$U, B, I = \text{Encoder}(\mathbf{X}, \mathbf{G}); \quad s_{ij} = \text{Predictor}(U_i, B_j) \quad (1)$$

where $U, B, I$ denote the learned embeddings of users, bundles and items respectively, and $s_{ij}$ is the predicted probability of user $i$ and bundle $j$ based on their embeddings. The loss objective $L_{tasks}(M_B)$ can be designed flexibly for various operations, e.g. contrastive learning [14], hard negative sampling [5]. Through the minimization of loss, the bundle recommender parameters are optimized.

4 METHODOLOGY

In this section, we introduce the proposed counterfactual learning paradigm to alleviate the problem of sparsity effectively in BR scenario. The complete process of our proposed Counterfactual Learning for Bundle Recommendation (CLBR) is presented in Algorithm 1, which consists of two main steps: counterfactual data augmentation with a heuristic node pairs sampler, and model training under counterfactual constraint.

4.1 Counterfactual Data Augmentation

In this part, we generate a set of counterfactual views based on the unified user-bundle-item relation graph as data augmentation. In detail, we use a sampler $S$ to choose user-bundle, user-item and bundle-item node pairs, and then change the structure of the original user-bundle-item relation graph by adding/dropping edges according to the sampling results. We set the augmentation ratios $r_{ui}, r_{ub}, r_{bi}$ for three types of node pairs respectively, to control the disturbance ratio of counterfactual views. We denote that $E^+_t$ is the set of node pairs selected to be added as edges to the counterfactual graph view, while $E^-_t$ is the set of edges picked for dropping, where $t$ denotes the type of node pairs which can be $ui, ub$ and $bi$.

There are many feasible methods to implement the sampler. In the simplest stochastic sampling method, we randomly select node pairs for $E^+_t$ from all unconnected node pairs in the original BR graph, as well as select node pairs for $E^-_t$ from existing edges in BR graph, until the size of $E^+_t$ and $E^-_t$ for each type satisfy the ratio we set. Formally, it can be expressed as $|E^+_t| \geq |E^-_t| \geq (1-\alpha)|E_t|$ for $\forall t$, where $\alpha \in [0, 1]$ controls the portion of adding and dropping. Then we can generate a counterfactual view of BR graph by applying these user-bundle, user-item and bundle-item edge adding and dropping operations to the original graph. Finally, we get a counterfactual view set through generating $N_c$ counterfactual views, which is a naive simulation of the counterfactual world.

The stochastic mechanism is easy to implement, but it brings too much noise to the generated graph views and reduces the effectiveness of subsequent model learning. So we designed a simple yet efficient heuristic sampling method in Algorithm 1, which improve the reliability of sampling by considering noise control. We pre-train a selection model $M_{th}$ which has the same structure of the recommendation model $M_B$, and utilize its graph encoder to generate all user, bundle and item embeddings $U, B, I$ before we
start the sampling phase. Considering the computational complexity, we randomly sample a batch of node pairs each time, query the embeddings of them, and calculate relevance scores for the node pairs in the batch. Taking user-bundle node pair \((i, j)\) as an example, the relevance scores can be calculated as \(r_{si,j} = U^T_i B_j\). Then we filter the noisy data by the following selection rule:

\[
\begin{align*}
E^+_{ub} & \leftarrow \bar{E}^+_{ub} \cup \{(i, j), \text{ if } r_{si,j} > \kappa^+ \text{ and } (i, j) \notin E_{ub}\} \\
E^-_{ub} & \leftarrow \bar{E}^-_{ub} \cup \{(i, j), \text{ if } r_{si,j} \leq \kappa^- \text{ and } (i, j) \in E_{ub}\}
\end{align*}
\]

No selection, else

where \(\kappa^+\) and \(\kappa^-\) are batch-based thresholds for selection, which are defined as the maximum and minimum relevance scores of user-bundle pairs in the batch multiplied by fixed ratios \(\alpha^+\) and \(\alpha^-\) respectively. This design simplifies the complex computation for full collection node pairs to the computation within every batch while ensuring noise filtering for effective augmentation. By repeating randomly sampling and selection process, we keep filling the adding set \(E^+\) and dropping set \(E^-\) until they contain user-bundle, user-item and bundle-item node pairs with total numbers of \(r_{ub}\), \(r_{ui}\) and \(r_{bi}\) respectively. Then we merge \(E^+\) into \(\bar{E}\) and remove \(E^-\) from \(\bar{E}\) for all \(t\) to generate a counterfactual view, and further get a counterfactual view set by generating \(N_u\) counterfactual views.

### 4.2 Counterfactual Constraint

Then the question we would like to answer is: given the augmented data composed of the original factual graph and generated counterfactual views, how to achieve effective learning for BR model \(M_0\)? Despite our noise control in the data augmentation process, some noise that deviates from the counterfactual distribution will inevitably remain in the generated data. In order to minimize the influence of residual noise, the model training needs to be additional constrained. Although the interaction data in the counterfactual world is different from that in the real world, user preferences, bundle and item features should be stable. While using GNN to perform semantic propagation on the counterfactual view and the original graph respectively, we expect that the embeddings of the same node learned on different graph views should be closer, while the distance between the node embedding distributions of the two views should vary within a reasonable range.

Therefore, for the normalized embeddings of user \(U\) and bundle \(B\), we have the following counterfactual constraints:

\[
\min \quad L_{\text{tasks}}(M_0)
\]

\[
\text{s.t.} \quad \sum_{i=1}^{N} D(U^+_i, U^+_j) \leq \lambda_u \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} D(U^+_i, U^+_j),
\]

\[
\sum_{i=1}^{K} D(B^+_{ci}, B^+_{cj}) \leq \lambda_b \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} D(B^+_{ci}, B^+_{cj}), \forall c \in \mathcal{C}
\]

where the function \(D(e_i, e_j)\) represents the distance between embeddings of node pair \(e_i\) and \(e_j\) in latent space, \(c\) denotes one potential counterfactual augmentation. The left consistency term in the two inequality reflects the consistency between the embeddings in the original graph and counterfactual view of the same user and bundle separately, and the right unrestraint term reflects the average consistency of every single node’s embedding on the original graph with the embeddings of all other nodes in the augmented counterfactual view. Compared to the constant threshold \(\lambda_u\), \(\lambda_b\) for the consistency between the real relation and counterfactual relation [26], we adopt the additional average consistency metrics for self-adaptive threshold settings. With lower average consistency, which can be explained as the user or bundle node being further from other nodes in latent space, we can make larger edge-level perturbations while ensuring that the node will not be confused with other nodes due to changes in embeddings caused by the perturbations. Through the above two constraints, the appropriate-intervention information is preserved and over-intervention information is discarded for exclusive user and bundle representations.

Directly solving the above constrained optimization problem is challenging for the constraints are not differentiable. Therefore we relax the constraints and convert the objective in Eq. 3 to the Lagrange optimization form as follows:

\[
\min \quad L = L_{\text{tasks}}(M_0) + \omega_u L_u + \omega_b L_b
\]

where non-negative hyper-parameter \(\omega_u\), \(\omega_b\) control the weight of the user- and bundle-side constraint loss \(L_u\), \(L_b\) respectively, which are defined as follows:

\[
L_u = \frac{1}{N} \sum_{i=1}^{N} \left( D(U^+_i, U^+_j) - \lambda_u \sum_{j=1, j \neq i}^{N} D(U^+_i, U^+_j) \right)
\]

\[
L_b = \frac{1}{K} \sum_{i=1}^{K} \left( D(B^+_{ci}, B^+_{cj}) - \lambda_b \sum_{j=1, j \neq i}^{K} D(B^+_{ci}, B^+_{cj}) \right)
\]

In our experiments, the distance metric \(D(e_i, e_j)\) is computed as \(-\exp(e_i^T e_j / \tau)\), where \(\tau\) denotes the temperature hyper-parameter for tuning how concentrated the features are in the latent space. In practice, \(D(e_i, e_j)\) can be any other distance metric function. To reduce computational complexity, the average global representations are not explicitly sampled but generated from the other user/bundle nodes within the same minibatch. And the final loss is computed across all representations of users/bundles in the minibatch.

### 5 FURTHER DISCUSSION

#### 5.1 Analysis about Noisy Control

The data augmentation process in CLBR depends on sampler \(S\). If \(S\) is not accurate, the augmented data can be noisy. To check the necessity of noise control in counterfactual data augmentation process, we would like to know how many data should be generated when given the noise level, if we want to ensure relatively good performance. So we theoretically analyze the relation between the amount of generated data and the noisy information introduced by \(S\), under the pursuit of well performance based on PAC learning framework[18]. We assume that our sampler can recover the true edges in the counterfactual graph views of the user-item-bundle graph with the probability of \(1 - \eta\), where \(\eta \in (0, 0.5)\) indicates the noisy level of \(S\). Then we have the following theory:

**Theorem 5.1.** Given a hypothesis class \(\mathcal{H}\), for any \(e, \delta \in (0, 1)\) and \(\eta \in (0, 0.5)\), if \(h \in \mathcal{H}\) is the edge ranking model learned based on the empirical risk minimization (ERM), and the sample complexity (i.e., number of samples) is larger than \(\frac{2\log(\frac{2M}{\delta})}{\epsilon^2(1-2\eta)^2}\), then the error between the model estimated and true results is smaller than \(\epsilon\) with probability larger than \(1 - \delta\).
The proof of this theorem can refer to [23]. Assuming that the generated data is noisy, if the prediction error of \( h \) is larger than \( \epsilon \), we have the empirical mismatching rate of \( h \) is smaller than \( \eta + \frac{e^{(1-2\gamma)}}{2} \), or the empirical matching rate of the optimal \( h^* \) is larger than \( \eta + \frac{e^{(1-2\gamma)}}{2} \). In order to achieve a good performance with given probability (i.e., \( \epsilon \) and \( \delta \)), at least \( \frac{2\log(\frac{2N}{\delta})}{e^{(1-2\gamma)}} \) samples are needed. That is to say, when the noise level of sampler \( \eta \) is larger, we need to choose much more edges for generating the counterfactual views, which is a huge computation consumption. This theory reveals the significance of our designed heuristic sampler with noise control. The adjustable threshold for selection rule roughly uniformly distributed [21, 28], then we have \( \lambda_i = 1 \). This means that the upper bound of \( H(U^f) \) is achieved if and only if \( \lambda_i = \frac{N}{2} \) for all \( i \). For \( L_{u_i} \), we expand it by Taylor series: \( L_{u_i} = \sum_{j=1}^{N} \sum_{k=1}^{N} U_{ij}^2 + \frac{1}{2} \sum_{j=1}^{N} U_{ij}^2 \). The \( L_{u_i} \) reaches the lower bound when \( \sum_{j=1}^{N} U_{ij} = 0 \). Hence, we have the empirical mismatching rate of \( \lambda_i = \frac{N}{2} \) for all \( i \), i.e., the learned embeddings are uniformly distributed. Here, \( \Sigma = \frac{N}{2} I \), thus the global optimum is exactly same: \( \lambda_i = \frac{N}{2} \). Hence we have lemma 2.

The two lemmas unveil the effects of two terms in our objective. Combining with above analysis, referring to [25, 28], we can further interpret Eq. 5 with the task-IRBR through following theorem:

**Theorem 5.4.** By optimizing \( L_u \), the task-relevant information \( I(U^f; T) \) is maximized, and the task-irrelevant information \( I(U^f; T) \) is minimized. Formally,

\[
\min_{\theta} L = \max_{\theta} I(U^f; T) \text{ and } \min_{\theta} I(U^f; T)
\]

Therefore, the learned representation of user and bundle is expected to contain minimal and sufficient information about BR. Through our counterfactual loss, the noised augmented data are filtered out efficiently and the task-relevant information is kept.

### 6 EXPERIMENTS

In this section, we conduct experiments on BR to evaluate the effectiveness of our method\(^1\). We aim to answer the following research questions:

**RQ1:** Does our proposed counterfactual learning paradigm contribute to better performance on existing BR models?  
**RQ2:** How do different designs in our paradigm influence the final performance?  
**RQ3:** Is the proposed paradigm sensitive to hyper-parameters? How do the hyper-parameters affect the effectiveness of our paradigm?

| Table 1: Statistics of datasets used in experiments. |
|---------------------------------------------------|
| **Statistic** | **NetEase** | **Youshu** |
| No. of items | 123,628 | 32,770 |
| No. of users | 18,528 | 8,039 |
| No. of bundles | 22,864 | 4,771 |
| No. of user-bundle | 302,303 | 51,377 |
| No. of user-item | 1,128,053 | 138,515 |
| No. of bundle-item | 1,778,838 | 176,667 |

### 6.1 Experimental Setup

**6.1.1 Dataset.** We conduct extensive experiments on two public datasets: NetEase and Youshu, which are widely used in the BR research [5, 6]. The statistics of all datasets after preprocessing are summarized in Table 1.

---

\(^1\)Our code and data will be released for research purpose.
Table 2: Experimental results (%) of models with different training paradigms in R@[20, 40], and NDCG@[20, 40] on two datasets. For all datasets, we perform 10-times to evaluate the performance, and report the mean and standard deviations. The bold number indicates the improvements over the related baseline are statistically significant (p < 0.01) with paired t-tests.

| Models       | NetEase          | Youshu           |
|--------------|------------------|------------------|
|              | R@20  | NDCG@20 | R@40  | NDCG@40 | R@20  | NDCG@20 | R@40  | NDCG@40 |
| BPRMF        | 3.512 ± 0.11     | 2.010 ± 0.10     | 2.617 ± 0.23 | 2.734 ± 0.20 | 19.63 ± 0.36 | 11.17 ± 0.25 | 27.35 ± 0.29 | 13.27 ± 0.19 |
| DAM          | 4.721 ± 0.21     | 2.404 ± 0.11     | 7.732 ± 0.34 | 3.210 ± 0.18 | 21.02 ± 0.38 | 12.05 ± 0.25 | 28.92 ± 0.47 | 14.10 ± 0.33 |
| RGCN         | 4.652 ± 0.23     | 2.370 ± 0.29     | 7.714 ± 0.45 | 3.170 ± 0.31 | 20.52 ± 0.39 | 11.52 ± 0.40 | 28.68 ± 0.48 | 13.67 ± 0.39 |
| RGCN + CASR  | 4.782 ± 0.25     | 2.437 ± 0.28     | 7.998 ± 0.35 | 3.192 ± 0.25 | 20.70 ± 0.38 | 11.63 ± 0.23 | 29.01 ± 0.42 | 13.82 ± 0.38 |
| RGCN + CLBR  | 5.210 ± 0.21     | 2.643 ± 0.23     | 8.494 ± 0.33 | 3.499 ± 0.19 | 22.17 ± 0.28 | 11.88 ± 0.23 | 31.05 ± 0.35 | 14.25 ± 0.25 |
| BundleNet    | 4.776 ± 0.41     | 2.545 ± 0.34     | 8.120 ± 0.41 | 3.225 ± 0.37 | 22.45 ± 0.47 | 12.00 ± 0.32 | 30.59 ± 0.25 | 14.19 ± 0.41 |
| BundleNet + CASR | 4.921 ± 0.33 | 2.583 ± 0.33     | 8.320 ± 0.39 | 3.402 ± 0.17 | 22.45 ± 0.48 | 12.03 ± 0.28 | 30.79 ± 0.24 | 14.34 ± 0.31 |
| BundleNet + CLBR | 5.332 ± 0.29  | 2.711 ± 0.24     | 8.730 ± 0.42 | 3.618 ± 0.28 | 22.53 ± 0.60 | 12.31 ± 0.27 | 32.11 ± 0.20 | 14.60 ± 0.35 |
| BGCN         | 5.760 ± 0.48     | 3.060 ± 0.28     | 9.020 ± 0.55 | 3.918 ± 0.33 | 22.69 ± 0.63 | 12.98 ± 0.43 | 31.07 ± 0.41 | 15.28 ± 0.38 |
| BGCN + CASR  | 5.911 ± 0.36     | 3.092 ± 0.28     | 9.315 ± 0.49 | 4.007 ± 0.34 | 22.70 ± 0.67 | 13.16 ± 0.40 | 31.87 ± 0.38 | 15.68 ± 0.32 |
| BGCN + CLBR  | 6.330 ± 0.31     | 3.298 ± 0.26     | 9.885 ± 0.30 | 4.255 ± 0.22 | 24.74 ± 0.63 | 14.10 ± 0.32 | 34.29 ± 0.27 | 16.73 ± 0.32 |
| CrossCBR     | 8.418 ± 0.05     | 4.565 ± 0.03     | 12.62 ± 0.06 | 5.689 ± 0.03 | 28.11 ± 0.06 | 16.68 ± 0.05 | 37.82 ± 0.05 | 19.37 ± 0.04 |
| CrossCBR + CASR | 8.423 ± 0.04 | 4.560 ± 0.02     | 12.66 ± 0.05 | 5.699 ± 0.03 | 28.16 ± 0.05 | 16.70 ± 0.05 | 37.85 ± 0.06 | 19.40 ± 0.03 |
| CrossCBR + CLBR | 8.721 ± 0.03  | 4.683 ± 0.03     | 12.85 ± 0.06 | 5.782 ± 0.04 | 28.48 ± 0.05 | 16.98 ± 0.04 | 38.30 ± 0.03 | 19.52 ± 0.04 |

- **NetEase** is a dataset collected by NetEase from its own music platform. Users can bind songs into a bundle or add bundles to their favorites on this platform.
- **Youshu** is collected by [4] from a book-review website Youshu, where every bundle is a list of books that users desired.

6.1.2 Baseline Models. To demonstrate the effectiveness of CLBR, we use following representative methods as baselines:

- **BPRMF** [15] uses a Bayesian Personalized Ranking pairwise learning framework to optimize the matrix factorization model.
- **RGCN** [17] is a classic GNN based model for heterogeneous graph, which consists of the multiple relations. Here we utilize it to model the user-item-bundle graph in BR.
- **DAM** [6] applies the attention mechanism multi-task learning to capture bundle-level association and collaborative signals.
- **BundleNet** [8] constructs a similar tripartite graph to well extract bundle representations from its included items’ features.
- **BGCN** [5] is a state-of-the-art model for BR that constructs bundle-level and item-level graphs and explicitly models the user-bundle interaction, user-item interaction and bundle-item affiliation by GNN.
- **CrossCBR** [14] is also a state-of-the-art model which utilizes contrastive learning to model the cooperative association between bundle-level and item-level graphs to improve BR.

Among these methods, BPRMF and DAM are the representatives of traditional models without using GNN, while the rest are graph-based methods. To verify the effectiveness of our proposed counterfactual learning paradigm, we trained the four graph-based models with different paradigms, then compare the recommendation result of every model. It is worth noting that for a fair comparison, the graph-based models with additional side-information are not considered in our BR experiments.

As our attempt to apply counterfactual thinking in BR domain is new, there are few suitable methods for comparison. So we choose counterfactual method from other recommendation domain that can be adapted to BR as a reference. Each baseline model is trained with the following counterfactual approaches:

- **CASR** [21] is a counterfactual data augmentation framework for sequential recommendation, which proposed a model-orient method to choose items with larger recommendation loss and generate counterfactual sequences. Here we leverage this principle to sample user-bundle pairs that provide larger recommendation loss and generate counterfactual graph views for BR.
- **CLBR** is our proposed counterfactual learning paradigm for bundle recommendation, which can refer to section 4.

6.1.3 Evaluation Metrics. To evaluate the recommendation performance, we employ two widely used metrics: Recall (R@k) and Normalized Discounted Cumulative Gain (NDCG@k) following [5, 8], where \( k = \{20, 40\} \).

6.1.4 Implementation Details. We implemented the baseline models and our proposed learning paradigm based on Pytorch. We generate four counterfactual views as the counterfactual view set \( G_{\text{CF}} \). The ratios \( r_{ui}, r_{ui}, r_{bi} \) for adding/dropping user-item, user-bundle and bundle-item edges are set as hyper-parameters, which can be adjustable for different models and datasets. The embedding dimension is set to 64. The temperature parameter \( \tau \) is set with a default value of 1. The ratios \( \alpha^2 \) and \( \alpha^-2 \) are set to 0.8 and 1.2. All parameters are initialized through a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. We employ the Adam optimizer to train the models with the mini-batch size 2048. We conduct the grid search over hyper-parameters as follows: learning rate in \( \{1e-5, 5e-5, 1e-4, 3e-4, 1e-3, 3e-3\} \), learning rate decay in \( \{0.01, 0.05, 0.1, 0.5\} \), learning rate decay step in \( \{2, 3, 4\} \), augmentation ratios in \( \{0.01, 0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\} \) and controlling factors \( \omega_{ui}, \omega_{bi} \) in \( \{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 30\} \). For each baseline, we adopt BPR loss and set the negative sampling rate to 1. The temperature factor \( \tau \) in counterfactual loss is set to 1. We tuned all the models through our best effort. Through 10 times experiments with
different run-time seed settings, we recorded all model’s average results as well as the fluctuation range of them.

6.2 Overall Comparison (RQ1)
We can obtain the following significant observations from the comparison results shown in Table 2.

Comparison of Different Baselines. While training the baselines under their original methods, we can observe that graph-based models generally achieve superior results, which proves the effectiveness of graph-based models for BR task. CrossCBR outperforms all baselines, demonstrating the effectiveness of applying graph contrastive learning in bundle recommendation. In other graph-based models, it should be noted that BGCN significantly perform better than BundleNet and RGCN. We attribute the reason to BGCN’s bundle-level and item-level graph modeling, which is better at differentiating users’ behavioral similarity and bundles’ content relatedness than BundleNet and RGCN’s user-bundle-item tripartite graph, and this design is also inherited by CrossCBR.

Observation of Paradigm Effectiveness. Compared with original models, the extended models with CLBR paradigm achieve significant performance improvement on all metrics consistently, while the improvement brought by the adjusted CASR is very limited, demonstrating the superiority of our method. For CLBR, the average improvements of each model on NetEase dataset are respectively 11.0%, 9.7%, 8.95%, 2.4% for RGCN, BundleNet, BGCN and CrossCBR while on Youshu dataset the improvements are 5.9%, 2.6%, 9.3%, 1.3%. The CLBR paradigm brings the underperforming model RGCN close to SOTA graph-based models. BGCN and CrossCBR have already obtained superior performance without data augmentation, but still can achieve vast performance improvement through our CLBR. These results demonstrate the effectiveness and generality of our learning paradigm on graph-based models for BR. We attribute these performance improvements to the combination of data augmentation and counterfactual constraint. After the data augmentation of CLBR, the data used for training has the potential to better represent the integral data distribution in the counterfactual world. Combined with the specially designed counterfactual loss, CLBR also achieves effective noise control during the model training step, so the models could learn more accurate and robust embeddings for users and bundles under limited training data. Therefore, the data sparsity problem in BR is alleviated, and the recommendations based on these embeddings show to be superior.

6.3 Ablation Study (RQ2)
Impact of Data Augmentation Methods. In this part, we compare different ways to generate the counterfactual views in CLBR, to check the impact of different data samplers and augmentation schemes. In more detail, we substitute sampling methods and augmentation schemes of CLBR paradigm, and conduct experiments on the SOTA model BGCN, which has vast improvement using CLBR. The method “CLBRSS” means the augmentation comes from a stochastic data sampler, while “CLBRMS” means the augmentation with heuristic data sampler proposed in Sec. 4.2, which is based on pretrained BR model $M_B$. For the data augmentation scheme, we compared the results of enriching the edges in BR graph individually for one relation or jointly for all relations. All hyper-parameters are tuned to their optimal value.

The results are presented in Table 3. We find that both the stochastic and heuristic sampler improve the models’ performance on most evaluation metrics, but the heuristic method is obviously more effective than the stochastic one. These results verify our conjecture about the disadvantage of stochastic mechanism in Sec. 4.2 for introducing too much noise in the process of generating enough counterfactual data, and prove the validness of heuristic sampler. Moreover, the joint augmentation for user-bundle, user-item and bundle-item relations further benefits models’ performance compared to enriching each type of data separately. We speculate that this joint augmentation can achieve a more comprehensive simulation of counterfactual distributions, which is necessary for BR.

Impact of Counterfactual Constraint Design. In this part, we aim to investigate the impact of counterfactual constraint designs. We compare the performance of BGCN model under three versions of CLBR paradigm with different counterfactual loss settings, including our proposed counterfactual loss with a self-adaptive threshold, constraint with a constant threshold, and no counterfactual loss. As shown in Table 4, we can find that the performances of counterfactual loss versions are higher than the no counterfactual version. By introducing the counterfactual loss, our framework can effectively filter out the remaining noise after augmentation. When comparing the two versions with counterfactual loss, we can observe that the version with self-adaptive threshold loss outperforms the constant threshold loss version consistently. This could be due to the fact that the self-adaptive threshold additionally considers the dynamic penalty term based on the distance between the local and global user/bundle representation distributions.

6.4 Hyper-parameters Study (RQ3)
Impact of Augmentation Ratios. For each augmentation ratio, we set the other two ratios and all remaining hyper-parameters to the optimal value that we’ve tuned in above experiments. Then we conduct experiments with varying the ratio from 0.01 to 0.3. We present the BGCN’s performance of Recall@20 on NetEase and Youshu datasets, respectively in Fig. 2. The results of other
Table 3: Experimental results (%) of two models with different augmentation settings in Recall@20, and NDCG@20 on two datasets. Here we omit "@20" on the evaluation metrics for simplicity due to space constraints.

| Methods | Augmentation Scheme | NetEase | BundleNet | Youshu | Youshu |
|---------|---------------------|---------|-----------|--------|--------|
|         | Recall | NDCG | Recall | NDCG | Recall | NDCG |
| CLBR_SS | UB only | 5.161 | 2.572 | 22.41 | 12.09 | 6.090 | 3.128 | 24.24 | 13.65 |
|         | UI only | 4.867 | 2.590 | 22.36 | 12.08 | 5.895 | 3.106 | 24.12 | 13.42 |
|         | BI only | 5.154 | 2.561 | 22.44 | 12.10 | 6.107 | 3.160 | 24.15 | 13.68 |
|         | UB-UI-BI | 5.212 | 2.682 | 22.46 | 12.17 | 6.266 | 3.241 | 24.60 | 13.99 |
| CLBR_MpS | UB only | 5.227 | 2.671 | 22.50 | 12.25 | 6.264 | 3.253 | 24.57 | 13.90 |
|         | UI only | 5.145 | 2.582 | 22.52 | 12.17 | 5.975 | 3.175 | 24.68 | 14.02 |
|         | BI only | 5.201 | 2.640 | 22.48 | 12.21 | 6.184 | 3.226 | 24.48 | 13.89 |
|         | UB-UI-BI | 5.332 | 2.711 | 22.53 | 12.31 | 6.330 | 3.298 | 24.74 | 14.10 |

Table 4: Experimental results (%) of BGCN with different counterfactual constraint designs in Recall@20 and NDCG@20 on two datasets. Here we omit "@20" on the evaluation metrics for simplicity due to space constraints.

| Constraint Design | NetEase | Youshu |
|-------------------|---------|--------|
|                   | Recall  | NDCG  | Recall | NDCG |
| w/o Counterfactual Loss | 5.922 | 3.094 | 23.36 | 13.21 |
| Constant threshold | 6.146 | 3.138 | 24.16 | 13.59 |
| Self-adaptive threshold | 6.330 | 3.298 | 24.74 | 14.10 |

models are omitted here for their similarity to BGCN’s and the space limitation. The polyline "UB" shows the experiment results of different $r_{ub}$ while $r_{ui}$ and $r_{bi}$ are fixed, and "UI" and "BI" likewise denote the results of various $r_{ui}$ and $r_{bi}$. The dotted line in figure presents the basic performance of BGCN without any augmentation. From Fig. 2 we can discover that the performances continue to rise until reaching the optimal augmentation ratios, then drop slowly as the ratios go up. We notice that the results remain stable when the augmentation ratios are around the optimal value, and even sub-optimal results still have considerable improvements compared to the base model. For example, on Youshu dataset, the results stabilize above 24.65 when tuning $r_{bi}$ in $[0.02, 0.05, 0.1]$, which are very close to the optimum. It reveals that the effectiveness of our CLBR is not sensitive to the augmentation ratios. In other words, we just need to adjust the augmentation ratios to a relatively loose range and then get nearly optimal results, which means the our CLBR could dramatically reduce the workload of parameter adjustment.

Impact of Controlling Factors. In Sec. 6.3, we have verified that counterfactual constraint will help improve graph-based models’ performance for BR. Here we want to further explore how the weights of constraint affect the performance of CLBR. For the controlling factors $\omega_u$ and $\omega_b$ of our proposed counterfactual loss, we fix all remaining hyper-parameters to the optimal value and conduct experiments. We show BGCN’s results using different factor combinations as 3D Surface in Figure 3. It can be observed that on NetEase and Youshu datasets, CLBR paradigm works well when both $\omega_u$ and $\omega_b$ are in a certain range from 0.01 to 1. The performance of BGCN in terms of Recall@20 is relatively stable when the controlling factors are not too large, as shown in the plateau of the figures. In other words, the CLBR paradigm is insensitive to $\omega_u$ and $\omega_b$ in a reasonable range, which demonstrates the robustness of it. But if we set either factor too large (e.g., > 5), the performance of the model is severely degraded, even much worse than the original version. This reveals that overlarge weight makes the counterfactual constraint dominate the total loss, thus the total loss deviates too much from the BR task, which leads to a drastic performance drop. So it is important to balance the weight of counterfactual loss by adjusting the parameters for different models and datasets.

6.5 Visualization and Analysis
As shown in Fig. 4, we utilize T-SNE [20] to visualize the embedding distribution of the users, bundles and items learned by BGCN on Youshu dataset, to find out how does CLBR paradigm affect representation learning in BR models. We use the "ori" to indicate the embeddings learned under original training paradigm, while "aug" means the embeddings are learned with augmented data under CLBR paradigm. The "init" or "prop" suffix indicates that the embeddings are obtained at initialization phase of the model or after the graph propagation. The nodes of user, item and bundle embeddings are colored green, purple and blue, respectively. We get the following observations: First, from the initialization case, compared with using original data, the embeddings initialized using augmented data evenly dispersed in representation space. Second, the graph propagation under the counterfactual constraint of CLBR paradigm can make the embeddings distribution become multiple uniform clusters, which is aligned with our expectation for optimal embeddings. These reveal that the augmented data has more potential to cover the counterfactual space, and the residual noise in the augmented data is effectively suppressed under our counterfactual constraints. Especially for users, the embeddings’ distribution of original data is in an extreme disequilibrium after initialization,
even graph propagation does not alleviate this situation well. But training with augmented data effectively solves this problem. It again demonstrates the superiority of our proposed CLBR paradigm.

7 CONCLUSION
In this paper, we pay special attention to the causal view on bundle recommendation, to find a solution for the data sparsity problem and address insufficient learning of representations to improve recommendation performance. Inspired by counterfactual thinking, we propose a novel graph-based counterfactual learning algorithm, including counterfactual data augmentation and counterfactual constraint. We apply the CLBR paradigm to several SOTA graph-based BR models, and conduct extensive experiments on real-world datasets to verify the effectiveness of it. As to future work, we would like to explore more interesting methods for counterfactual data augmentation. It’s also meaningful to generalize this paradigm into more recommendation fields.

REFERENCES
[1] Ehsan Abbassnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. 2020. Counterfactual vision and language learning. In CVPR. 10044–10054.
[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. arXiv preprint arXiv:1706.02063 (2017).
[3] Da Cao, Liqiang Nie, Xiangnan He, Xiaochi Wei, Shunzhi Zhu, and Tat-Seng Chua. 2017. 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. 585–594.
[4] Da Cao, Liqiang Nie, Xiangnan He, Xiaochi Wei, Shunzhi Zhu, and Tat-Seng Chua. 2017. Embedding factorization models for jointly recommending items and user generated lists. In SIGIR. 585–594.
[5] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Bundle recommendation with graph convolutional networks. In SIGIR. 1673–1676.
[6] Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng. 2019. Matching User with Item Set: Collaborative Bundle Recommendation with Deep Attention Network. In IJCAI. 2095–2101.
[7] Qilin Deng, Kai Wang, Minghao Zhao, Runze Wu, Yu Ding, Zhenhe Zou, Yue Shang, Jiaorong Tao, and Changjie Fan. 2021. Build Your Own Bundle - A Neural Combinatorial Optimization Method.
[8] Qilin Deng, Kai Wang, Minghao Zhao, Zhenhe Zou, Runze Wu, Jiaorong Tao, Changjie Fan, and Liang Chen. 2020. Personalized bundle recommendation in online games. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2381–2388.
[9] Yash Goyal, Ziyun Wu, Jan Ernst, Dhrav Batra, Devi Parikh, and Stefano Lee. 2019. Counterfactual Visual Explanations. In IJMLC. Vol. 97. 2376–2384.
[10] Miha Glicic, Dunja Madanic, Blaž Fortuna, and Marko Grobelnik. 2005. Data sparsity issues in the collaborative filtering framework. In International workshop on knowledge discovery on the web. Springer, 58–76.
[11] Xingchen Li, Xiang Wang, Xiangnan He, Leng Chen, Jun Xiao, and Tat-Seng Chua. 2020. Hierarchical fashion graph network for personalized outfit recommendation. In SIGIR. 159–168.
[12] Guangnan Liu, Yanjie Fu, Guoqing Chen, Hui Xiong, and Can Chen. 2017. Modeling buying motives for personalized product bundle recommendation. TKDD 11, 3 (2017). 1–26.
[13] Yidan Liu, Min Xie, and Laks VS Lakshmanan. 2014. Recommending user generated item lists. In RecSys. 185–192.
[14] Yunshan Ma, Yingzhe He, An Zhang, Xiang Wang, and Tat-Seng Chua. 2022. CrossCBR: Cross-view Contrastive Learning for Bundle Recommendation. arXiv preprint arXiv:2206.00242 (2022).
[15] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012).
[16] Oren Sar Shalom, Noam Koernigstein, Ulrich Paquet, and Hastagiri P Vanchinathan. 2016. Beyond collaborative filtering: The list recommendation problem. In WWW. 63–72.
[17] Michael Schlichtkrull, Thomas N Kipf, Peter Blom, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In European semantic web conference. Springer, 593–607.
[18] Shai Shalev-Shwartz and Shai Ben-David. 2014. Understanding machine learning: From theory to algorithms. Cambridge university press.