Robust Basket Recommendation via Noise-tolerated Graph Contrastive Learning

Xinrui He*
University of Illinois at Urbana
Champaign
USA
xhe33@illinois.edu

Tianxin Wei*
University of Illinois at Urbana
Champaign
USA
twei10@illinois.edu

Jingrui He
University of Illinois at Urbana
Champaign
USA
jingrui@illinois.edu

ABSTRACT

The growth of e-commerce has seen a surge in popularity of platforms like Amazon, eBay, and Taobao. This has given rise to a unique shopping behavior involving baskets – sets of items purchased together. As a less studied interaction mode in the community, the question of how should shopping basket complement personalized recommendation systems remains under-explored. While previous attempts focused on jointly modeling user purchases and baskets, the distinct semantic nature of these elements can introduce noise when directly integrated. This noise negatively impacts the model's performance, further exacerbated by significant noise (e.g., a user is misled to click an item or recognizes it as uninteresting after consuming it) within both user and basket behaviors. In order to cope with the above difficulties, we propose a novel **B**asket recommendation framework via **N**oise-tolerated **C**ontrastive **L**earning, named BNCL, to handle the noise existing in the cross-behavior integration and within-behavior modeling. First, we represent the basket-item interactions as the hypergraph to model the complex basket behavior, where all items appearing in the same basket are treated as a single hyperedge. Second, cross-behavior contrastive learning is designed to suppress the noise during the fusion of diverse behaviors. Next, to further inhibit the within-behavior noise of the user and basket interactions, we propose to exploit invariant properties of the recommenders w.r.t augmentations through within-behavior contrastive learning. A novel consistency-aware augmentation approach is further designed to better identify the noisy interactions with the consideration of the above two types of interactions. Our framework BNCL offers a generic training paradigm that is applicable to different backbones. Extensive experiments on three shopping transaction datasets verify the effectiveness of our proposed method.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Basket Recommendation, Graph Contrastive learning, Denoising

1 INTRODUCTION

Recommender systems have become a powerful tool that greatly enhances shopping experiences on online platforms ever since their inception [1, 4]. In practice, customers often purchase multiple items at the same time, and the co-occurrence relationships could provide rich information in item properties mining. Basket recommendation [39] is to predict a set of relevant items that a customer will be interested in by analyzing the composition of the historical interactions and the current shopping baskets if given [1], which can be better used for product arrangement, procurement, promotion, and marketing [12] to improve customer experience and generate business value.

There are an increasing number of works [5, 14, 25] trying to explore the customer's order history to capture the users' shopping preference and item semantics, with the aim of improving the recommendation quality and boosting online service. However, not all of the order details in the transaction data are essential and relevant to determining the user's next action. There are usually some user-item interactions as well as basket-item interactions appearing as noise due to the diversity of the basket contents and the users' mismatched behaviors. In practice, the noisy interaction occurs in the shopping transaction data for the reason that the users' shopping behavior is somewhat random and fragmented, meaning that users sometimes buy items against their shopping habits or there are some items in the basket that are not related to any others. The existence of such noisy interactions is verified [33, 49, 51] to hinder the understanding of the user's behavior patterns, which will harm the recommender system training and thus hinder its practical deployment.

Therefore, it is necessary to denoise in basket recommendation (BR) to extract effective information to enhance recommendation performance. However, most existing BR methods [3, 24, 41] mainly tend to jointly model purchases and baskets which will introduce noise to the learned representations due to the heterogeneity of user

---

[1]A customer may go to this supermarket to buy products many times during a period. Then this customer has multiple shopping baskets
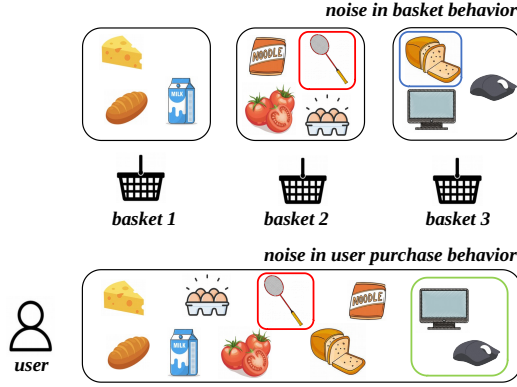
Figure 1: An example of the noise in the basket recommendation: in terms of content relevance in the basket, the food in basket 3 is the noisy item; from the perspective of the users' shopping preferences, it is the electronics in basket 3 that might be treated as the noise. From both two perspectives, the racket shows up as a consistent noise.

purchases and basket behaviors. To suppress the noise in recommender systems, current works can be classified as follows: (i) The first line of work [30, 49, 51] focuses on improving model robustness [38] against user interaction noises.; (ii) Another emerging series of papers [23, 33] put their attention on inhibiting the negative effect of noisy basket behaviors. Both the two types of works only consider the noise within one behavioral pattern, while the joint noise handling of the above two behaviors remains unexplored. What's more, the semantic mismatch of user purchases and basket behaviors will introduce additional noise during the behavioral fusion process. Consequently, it's essential to design the denoise method from a global view that considers the comprehensive information from both two behaviors. To illustrate the noise, we also give an example in Figure 1, which presents the consuming history of a user from the view of user purchase behavior (below) and the view of basket behavior (above). We observe that, from the basket behavior view, the bread in the basket 3 can be regarded the outlier with high probability. However, if we take a look at the user purchase behavior, the bread could not be treated as an anomaly while the computer and mouse are more likely to noise for the user. Moreover, if we look at the racket, which is both unimportant in the two views, there is a high probability for it to be a consistent noise. From the example, we're inspired that it's crucial to design the denoise approach that considers information comprehensively from both within and across the two behavior views.

In this paper, we propose a comprehensive within-basket recommendation framework BNCL via noise-tolerated contrastive learning to handle the noise existing in the cross-behavior integration and within-behavior modeling. To be more specific, first, we adopt the typical user-item bipartite graph to model the user-item interactions from the user purchase behavior, and the basket-item interactions for the basket behavior are represented as the hypergraph to model the complex basket behavior, where all the items appearing in the same basket are treated as a single hyperedge. Secondly, we propose cross-behavior contrastive learning to fuse the representations

learned from the basket behavior into recommender systems, which aims at suppressing the noise introduced by the fusion of diverse behaviors. Then, to handle the within-behavior noise of the user and basket interactions respectively, we propose to exploit invariant properties of the recommenders w.r.t augmentations through within-behavior contrastive learning. During the process, a novel consistency-aware augmentation method is proposed to better identify the noisy interactions with the comprehensive information of both the two types of behaviors. To optimize the model, we leverage a multi-task training strategy to jointly optimize the classic recommendation task and the self-supervised contrastive denoising task. In summary, the contributions of this paper could be summarized as follows:

- This work formulates the idea of integrating the basket behaviors into user-item interaction modeling with a light hypergraph message passing schema as well as a joint self-supervised learning paradigm.
- We systematically illustrate the noise issues in the within-basket recommendation problem and propose a general basket recommendation framework BNCL to improve robustness against the within-behavior and cross-behavior noise issues via noise-tolerated contrastive learning.
- Extensive experimental results over three shopping transaction datasets show that our proposed method outperforms state-of-the-art baselines in terms of various ranking metrics.

The rest of the paper is organized as follows. We show the preliminary definition in Section 2 and introduce the proposed BNCL in Section 3. Then we present the experimental results in Section 4. Section 5 briefly discusses the existing work. In the end, we conclude the paper in Section 6.

2 PRELIMINARY

2.1 Within-basket Recommendation Setting

As a common practice, we use $U = \{u_1, u_2, \ldots, u_{|U|}\}$ to represent all users and $I = \{i_1, i_2, \ldots, i_{|I|}\}$ to represent all items where $|U|$ and $|I|$ denote the number of the users and the items respectively. We consider the basket as a set of items that the user ordered in one transaction. Therefore, we will obtain an interaction basket sequence according to the transaction record of the user u, which is denoted as $B^u = \left(b_1^u, b_2^u, \ldots, b_{|B^u|}^u\right)$ where $|B^u|$ is the number of the baskets that user u has purchased and $b_j^u \subseteq I$ represents the jth basket purchased by user u. Within-basket recommendation task aims to recommend the most possible item list to be added to a partially given basket b_p^u associated with a user u.

2.2 User-item Interaction View Learning

In the raw transaction data, each user and item is assigned a unique ID respectively. We use a d dimension embedding to represent each user and item where $e_u \in \mathbb{R}^d$ is the embedding of the user, $e_i \in \mathbb{R}^d$ is the embedding of the item. To capture the users' purchase behavior, we adopt the idea of the user-item bipartite graph to model the user-item interactions, in which the user and the item are treated as nodes. If the user u has bought the item i, there will be an edge connecting these two nodes on the user-item interaction graph.
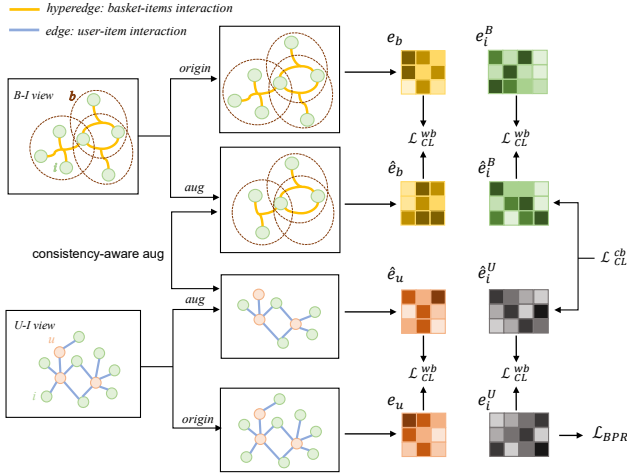
Figure 2: The overall framework of BNCL. **(i) In the view of basket behavior, the yellow hyperedges denote the baskets and the nodes denote the items. In the user purchase behavior, the blue edges represent the user-item interactions. (ii) We adopt within-basket contrastive learning between the augmented graph and the original graph for this two behavior respectively. (iii) The augmented views of the user purchase behavior and the basket behavior are obtained by the proposed consistency-aware augmentation. (iv) Cross-behavior contrastive learning helps to better fuse the item properties learned from two behaviors.**

We adopted the graph convolutional network LightGCN [17] to perform the message passing on the user interaction graph.

Following LightGCN, the k-th layer information propagation on the user-item bipartite graph could be described as follow:

$$\begin{cases} \mathbf{e}_u^{(k)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{U(k-1)} \\ \mathbf{e}_i^{U(k)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(k-1)} \end{cases} \quad (1)$$

where $\mathbf{e}_u^{(k)}, \mathbf{e}_i^{U(k)} \in \mathbb{R}^d$ are the embedding of the user u and the item i at k-th layer respectively. We randomly initialize the user and item embedding $\mathbf{e}_u^{(0)}, \mathbf{e}_i^{U(0)}$ at the very beginning.

To take the aggregating information at different depths into account, the final representations $\mathbf{e}_u$ and $\mathbf{e}_i^U$ of the user and the item are obtained as the mean of the output embedding of different layers:

$$\mathbf{e}_u = \frac{1}{K} \sum_{k=0}^{K} \mathbf{e}_u^k \quad (2)$$

$$\mathbf{e}_i^U = \frac{1}{K} \sum_{k=0}^{K} \mathbf{e}_i^{U(k)} \quad (3)$$

where K is the number of the layers, $\mathbf{e}_u$ and $\mathbf{e}_i$ are the final representations of the user and the item.

2.3 Contrastive learning

Intuitively, the different views of the same sample are likely to be clustered in the embedding space. Contrastive learning (CL) aims at

learning good data representations by reducing the distance of the defined positive pairs while pushing away the representations of the negative pairs in the embedding space. The main process of CL [42] is to first construct the diverse views of the raw data through an augmentation set $\mathcal{A}$. Given a data point x, the augmentation results of x are denoted as $\mathcal{A}(x)$, which are treated as the positive pairs, otherwise, the negative pairs. CL tries to learn an encoder f such that the positive pairs are well aligned and the negative ones have been pushed apart. We denote the $z(x)$ as the representation of sample x in the embedding space. To achieve this goal, a class of methods employed InfoNCE [7, 9, 32] as the contrastive loss function, formulated as:

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{x \in \mathcal{X}, x' \in \mathcal{A}(x)} -\log \frac{\exp\left(\text{sim}\left(x, x'\right)/\tau\right)}{\sum_{x_k \in \mathcal{X}} \exp\left(\text{sim}\left(x, x_k\right)/\tau\right)} \quad (4)$$

where $\mathcal{X}$ is the set of data samples in a mini-batch, x' is the augmented view of a random data point x, $\text{sim}(\cdot, \cdot)$ is a similarity function and τ is the temperature coefficient used to control the uniformity of the representation in the embedding space. In this case, the model is more likely to learn the more invariant and essential properties of the raw data by mapping the positive pairs into the nearby space.

To adapt to the downstream task, it is natural to adopt the contrastive objective as a ladder combined with supervised signals to form a multitask learning objective.

3 METHOD

In this section, we introduce the proposed BNCL. The overall framework is shown in Figure 2. First, we model the user-item and basket-item interactions from the user purchase behavior and the basket behavior through a user-item interaction graph and a complementary basket hypergraph respectively. Then, we propose cross-behavior contrastive learning to eliminate the noise introduced in the fusion of the representations learned from purchase and basket behavior. Finally, within-behavior contrastive learning is employed in the user's purchase behavior and the basket behavior respectively for denoising these two behaviors. In addition, a consistency-aware augmentation is designed in the within-behavior contrastive learning to help identify the true noisy interactions.

3.1 Complementary View Learning

In the basket recommendation task, the orders are usually unique to the users, which means it is not likely that two customers share exactly the same basket. Compared with user-basket interactions, what matters most is the interactions between users and items, which contains the cross-basket information, as well as the interrelated information among items within a basket. Thus, we propose to learn the representations from both the user purchase behavior and the basket behavior.

For the user purchase behavior, which contains the user-item interactions, we employ the user-item interaction graph and perform LightGCN message passing on it to learn the representations of the user and item, denoted as $\mathbf{e}_u$ and $\mathbf{e}_i$. From the user purchase behavior, we encode the user's personalized information and shopping preferences to the user and item embedding.

Different from the general recommendation task, the basket recommendation scenario is provided with the shopping basket records which contain valuable correlation information about the items. Specifically, the compositions in a basket often imply the complementarity, similarity, or substitution relationships among the item. The basket behavior is encoded in the basket-item interactions, which serves as a supplementary part to extract effective item properties for recommendation. We use a basket hypergraph $\mathcal{G}_{hyper} = (V, E)$ to model the basket behavior for the following reasons. (i) In the hypergraph, a hyperedge can connect two or more vertices [59], which can well model the diverse relationship between a basket and multiple items. (ii) The hypergraph has the ability to aggregate high-order information through the message passing among hypernodes and hyperedges, which is essential to capturing the complex relations between various baskets and items. Here V is the set of vertices and each vertice represents an item in the item set. We use ϵ to denote the basket-item interactions, which means that a hyperedge $\epsilon \in E$ denotes a basket b and the vertices it connects represent all the items belonging to this basket. The basket hypergraph contains $|I|$ vertices and M hyperedges. The relationship between vertices and hyperedges can be described by an incidence matrix $H \in \mathbb{R}^{|I| \times M}$ defined as follows:

$$\mathbf{H}(v, \epsilon) = \begin{cases} 1, & \text{if } \epsilon \in E \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Following the spectral hypergraph convolution proposed in [11], we design light message passing on the constructed hypergraph to effectively perform the information aggregation with hyperedges as the mediators. The representation of vertices $E_I^{B(k)}$ at k^{th} layer is obtained as :

$$\mathbf{E}_I^{B(k)} = \mathbf{D}^{-1/2} \mathbf{H} \mathbf{B}^{-1} \mathbf{H}^T \mathbf{D}^{-1/2} \mathbf{E}_I^{B(k-1)} \tag{6}$$

where $D \in \mathbb{R}^{|I| \times |I|}$ and $B \in \mathbb{R}^{M \times M}$ are the diagonal degree matrices of the vertices and the hyperedges, where each entry denotes the degree value of corresponding items/hyperedges. $E_I^{B(k)} \in \mathbb{R}^{|I| \times d^k}$ is the item embedding matrix at the kth layer on the hypergraph and d^k is the hidden size of kth layer. We use $e_i^{B(k)}$ to denote the representation of item i at the kth layer, which is equal to the ith row of $E_I^{B(k)}$.

We use the mean of the representations at different layers as the final representation of the items learned from the basket behavior:

$$\mathbf{e}_i^B = \frac{1}{K} \sum_{k=0}^{K} \mathbf{e}_i^{B(k)} \tag{7}$$

Defining $\mathcal{N}_b = \{i \mid \mathbf{H}(i, b) = 1\}$ as the items incorporated in the basket b, we can obtain the final basket embedding from the items:

$$\mathbf{e}_b = \frac{1}{|\mathcal{N}_b|} \sum_{i \in \mathcal{N}_b} \mathbf{e}_i^B \tag{8}$$

3.2 Cross-behavior Contrastive Learning

Item embedding e_i^U and e_i^B, learned from the user purchase behavior and the basket behavior, capture inherent semantic properties of different views. However, loosely integrating them together may lead to suboptimal results due to the inconsistent feature space and view aggregation schema. To accomplish the fusion of diverse

behaviors, in this section, we propose cross-behavior contrastive learning to suppress the noise during the fusion of the user purchase behavior and the basket behavior.

Motivated by the core idea of CL, we introduce cross-behavior contrastive learning to align the representation of the same item generated from different behaviors, which could unify the item embedding into the same feature space and further explore the intrinsic semantic properties of the item.

The representation of the same item should share the same intrinsic semantic properties. Thus, we define the positive pairs as the embedding of the same item learned from the augmentation views of the user purchase behavior and the basket behavior, denoted as $\hat{e}_i^U$ and $\hat{e}_i^B$. Then we treat the representations of different item embedding learned from different behaviors as negative pairs. Employing the InfoNCE [32], the objective of cross-behavior contrastive learning could be formulated as:

$$\mathcal{L}_{CL}^{cb} = \sum_{i \in I} -\log \frac{\exp\left(\text{sim}\left(\hat{e}_i^U, \hat{e}_i^B\right)/\tau\right)}{\sum_{j \in I, i \neq j} \exp\left(\text{sim}\left(\hat{e}_i^U, \hat{e}_j^B\right)/\tau\right)} \tag{9}$$

where $\text{sim}(\cdot, \cdot)$ is cosine similarity function, τ is the temperature of the contrastive learning.

3.3 Within-behavior Contrastive Learning

In the above section, we model the user purchase behavior and the basket behavior through a user-item interaction graph and a basket hypergraph and learn the representation from these two graphs. However, in real-world shopping transaction records, there are always some items that are irrelevant to the user preference or to the intents of the basket which appear as noise. The existence of noise will affect the learned user shopping preferences and deteriorate the quality of the representation.

To tackle this problem, we propose within-behavior contrastive learning on the user purchase behavior and the basket behavior respectively to achieve the goal of denoising the user-item interactions and the basket-item interactions. To be more specific, we generate the augmented view of the user-item interaction graph and the basket hypergraph. Then, the message passing is performed both on the original graph and the augmented graph. We denote the representation of the user and item on the user-item interaction graph as e_u and e_i^U, the representation of the basket and item on the basket hypergraph as e_b and e_i^B. Also, we have the representation of the user and item $\hat{e}_u$, $\hat{e}_i^U$ on the augmented view of the user-item interaction graph, and the representation of the basket and item $\hat{e}_b$, $\hat{e}_i^B$ on the augmented view of the basket hypergraph. The objective of the user representation learning in within-behavior contrastive learning is defined as:

$$\mathcal{L}_{CL}^U = \sum_{u \in \mathcal{U}} -\log \frac{\exp\left(\text{sim}\left(e_u, \hat{e}_u\right)/\tau\right)}{\sum_{v \in U, u \neq v} \exp\left(\text{sim}\left(e_u, \hat{e}_v\right)/\tau\right)} \tag{10}$$

We adopt the same contrastive objectives for item embedding of the original view and the augmented view on the user-item interaction graph, and the basket embedding as well as the item embedding of the original view and the augmented view on the basket hypergraph. Four contrastive terms are obtained as $\mathcal{L}_{CL}^U$,

$\mathcal{L}_{CL}^{I_u}$, $\mathcal{L}_{CL}^{B}$, $\mathcal{L}_{CL}^{I_b}$. The final contrastive learning loss for within-behavior denoising is the sum of these four terms:

$$\mathcal{L}_{CL}^{wb} = \mathcal{L}_{CL}^{U} + \mathcal{L}_{CL}^{I_u} + \mathcal{L}_{CL}^{B} + \mathcal{L}_{CL}^{I_b} \qquad (11)$$

3.4 Consistency-aware Augmentation

Up to now, we have investigated denoising techniques applicable for each behavior respectively, as well as during the fusion of heterogeneous behaviors. However, developing an effective augmentation strategy for contrastive learning that precisely identifies the noisy interactions, considering these two types of behaviors, remains an unresolved challenge.

Commonly used data augmentation methods for graph structures, such as stochastic edge and node removal, introduce substantial randomness in altering the graph's structure and identifying noise. For example, when the edges removed are connected to important nodes, some fundamental relationships may be lost and the underlying structure of the graph may be destroyed as a result. Moreover, previous augmentation techniques that rely on a single view can be unreliable in the context of basket recommendation. An item can be considered noise from one perspective, while it may carry meaningful information when viewed from another perspective, as depicted in Figure 1. To mitigate the risk of losing important relationships, it is crucial to develop an augmentation strategy that integrates crucial multi-view information and removes interactions that consistently exhibit noise.

Aiming to identify the noise, we propose a consistency-aware augmentation approach for cross-behavior and within-behavior contrastive learning. To be more specific, we comment that a user-item interaction or a basket-item interaction tends to be noisy when the item is considered potentially noisy based on both the user purchase behavior and the basket behavior. Inspired by the definition of the node centrality in the graph [31], we define the interaction importance s_{u-i}^{ϵ} and s_{b-i}^{ϵ} upon two graphs as:

$$s_{u-i}^{\epsilon} = \log\left(\delta_d(u) + \delta_d(i) + \delta_{d\_hyper}(i)\right) \qquad (12)$$

$$s_{b-i}^{\epsilon} = \log\left(\delta_{d\_hyper}(b) + \delta_{d\_hyper}(i) + \delta_d(i)\right) \qquad (13)$$

where $\delta_d(u)$ and $\delta_d(i)$ are the degree of the user and item node on the user-item interaction graph. On the basket hypergraph, we focus on both the degree of the hyperedge $\delta_{d\_hyper}(b)$ which is defined as the number of items contained on the basket hyperedge, and the degree of the item vertex $\delta_{d\_hyper}(i)$ which is defined as the number of hyperedges connecting to this item vertex.

According to the expression of edge importance, the importance of an edge is defined from the perspective of the user purchase behavior as well as the basket behavior. We comment that a user-item interaction or a basket-item interaction is more important when it has greater edge importance on the corresponding graph. To generate more meaningful augmentation, we are inclined to drop the less important interactions.

To calculate the probability of dropping an edge ϵ, for example, on the user-item interaction graph, we use normalization to transform the edge importance s_{u-i}^{ϵ} of edge ϵ into the probability:

$$p_{\epsilon} = \frac{s_{\max} - s_{u-i}^{\epsilon}}{s_{\max} - s_{\min}} \cdot p \qquad (14)$$

Table 1: Dataset Statistics.

	#Users	#Items	Average Basket Size	Average #Baskets per User
Instacart	22168	40044	37.00	2.96
Tafeng	7119	11916	15.99	2.20
Valuedshoppers	9532	7860	19.94	17.66

where p is the overall edge drop probability, $s_{\max}$ and $s_{\min}$ is the max and min value of s_{u-i}. Following the same formula, the probability of dropping a basket-item interaction ϵ on the basket hypergraph can be obtained as well. According to the probability p_{ϵ} of dropping the edge ϵ, we could obtain the augmentation view of the original graph $\mathcal{G}$ where the probability that the edge ϵ belongs to the augmentation view $\hat{\mathcal{G}}$ is $1 - p_{\epsilon}$.

With consistency-aware augmentation, the positive pairs generated from two behavior views tend to share more common intrinsic properties for better alignment in cross-behavior contrastive learning. While it can generate the more meaningful augmentation view for within-behavior contrastive learning on the user-item interaction graph and basket hypergraph respectively.

3.5 Prediction and Optimization

To perform recommendations, the ranking score of each user-item pair considers both the user information and current basket information:

$$\hat{y}(u, b, i) = (1 - r) \cdot \mathbf{e}_u^{\top} \mathbf{e}_i + r \cdot \mathbf{e}_b^{\top} \mathbf{e}_i \qquad (15)$$

where $\mathbf{e}_i = \mathbf{e}_i^U + \mathbf{e}_i^B$ denotes the fused item embedding performed with contrastive denoising, r is a hyperparameter to balance the capacity of the user and the basket in the recommendation.

We use the BPR loss [34] as the main recommendation loss. We sample a positive item i and a negative item j for the user u, where the positive item is selected within the current basket and the negative item is sampled from items without being purchased. The main loss is:

$$L_{\text{main}} = -\sum_{(u,i,j)} \log \sigma\left(\hat{y}_{(u,b,i)} - \hat{y}_{(u,b,j)}\right) + \lambda \|\Theta\|_2^2 \qquad (16)$$

where Θ is the model parameters and λ is a positive constant. To improve recommendation with the self-supervised denoising tasks, we leverage a multi-task training strategy to jointly optimize the classic recommendation task, the cross-behavior contrastive task and the within-behavior contrastive task. In this way, we have the final multi-task loss as:

$$\mathcal{L} = \mathcal{L}_{main} + \alpha_1 \mathcal{L}_{CL}^{cb} + \alpha_2 \mathcal{L}_{CL}^{wb} \qquad (17)$$

where α_1 and α_2 are hyperparameters to control the linear weight.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of our proposed BNCL. Our experiments intend to answer the following research questions:

- **RQ1:** How does BNCL perform in the within-basket recommendation task compared with the baseline models?

Table 2: Experimental results on the three real-world datasets through different methods with % omitted. The best results are highlighted in boldface. Underlined values indicate the second best.

Data	Metric	Simple methods	MF-based methods		NBR methods		GNN-based methods			Denoising methods		BNCL
		PersonPop-k	BPRMF	Triple2Vec	DREAM	TIFU-KNN	LightGCN	Basconv	MITGNN	SGL	CLEA	
Instacart	Recall@40	1.959	12.005	9.285	9.430	13.567	13.792	12.569	11.461	15.742	_15.919_	**17.156**
	Recall@60	2.309	15.612	11.711	11.105	20.407	17.832	17.144	15.426	19.350	_20.701_	**21.099**
	HR@40	13.377	56.928	46.405	44.542	54.174	61.355	57.725	64.705	63.423	_65.334_	**67.169**
	HR@60	15.433	66.048	53.930	49.553	74.197	70.313	68.690	64.985	70.286	_74.356_	**74.674**
	NDCG@40	4.127	14.451	14.359	12.061	5.794	18.463	15.157	14.526	21.492	_19.390_	**22.742**
	NDCG@60	4.541	19.876	16.474	13.440	9.263	21.504	18.581	17.630	24.162	_22.760_	**25.615**
TaFeng	Recall@40	0.263	2.075	0.082	0.381	2.459	3.178	2.799	2.745	3.283	_3.382_	**3.488**
	Recall@60	0.468	2.648	0.188	0.513	2.836	3.781	3.437	3.467	3.893	_4.062_	**4.175**
	HR@40	1.063	6.634	0.300	1.634	8.586	10.200	9.052	8.867	10.583	_10.755_	**11.845**
	HR@60	1.905	8.369	0.612	2.177	9.470	11.948	10.966	10.934	12.356	_12.607_	**12.835**
	NDCG@40	0.234	2.051	0.078	0.321	0.870	3.898	2.879	2.497	4.000	_4.054_	**4.356**
	NDCG@60	0.389	2.378	0.134	0.413	0.978	4.244	3.257	2.902	4.351	_4.428_	**4.628**
ValuedShopper	Recall@40	1.063	1.963	1.655	1.638	_6.567_	5.844	5.476	5.903	5.995	5.384	**6.659**
	Recall@60	1.486	2.873	2.856	1.979	_8.206_	7.780	7.281	7.619	8.045	7.505	**8.845**
	HR@40	3.910	6.359	5.417	3.161	20.219	19.600	18.413	_20.275_	20.216	17.701	**22.136**
	HR@60	5.494	9.324	9.417	4.336	_26.521_	25.221	23.690	25.234	26.224	23.822	**28.334**
	NDCG@40	1.092	1.840	1.747	1.579	2.581	5.489	5.485	_6.035_	5.659	4.766	**6.364**
	NDCG@60	1.374	2.408	2.505	1.788	3.253	6.634	6.572	_7.153_	6.882	6.008	**7.529**

- **RQ2:** How do different components in BNCL contribute to the performance?
- **RQ3:** How is the generalization ability of our proposed BNCL under different circumstances (e.g., varying length of recommended item list and backbones)?
- **RQ4:** How does the proposed BNCL perform in the presence of noise (the robustness of BNCL to the varying ratio of noise added)?

4.1 Dataset

We evaluate the within-basket recommendation performance on real-world datasets: Instacart [2], Tafeng [3] and Valuedshoppers [4].

- **Instacart** is a transaction dataset collected from an online shopping grocery. It contains the record of over 3 million grocery orders over time which come from more than 200,000 users.
- **Tafeng** contains Chinese grocery store transaction data over four months released by ACM RecSys. It consists of the records of over 13000 users' shopping orders.
- **Valuedshoppers** provides almost 350 million purchase histories from over 300,000 shoppers which includes a large set of users' basket-level shopping behaviors. Considering a large number of records, We sampled the transactions for training and prediction.

We treat the set of items purchased by a user during a time session as a shopping basket. In order to make the basket informative enough to be useful in the algorithm, we remove baskets containing less than 30 items for Instacart, and less than 10 items for the Valuedshoppers and Tafeng due to the sparsity of the basket-item

interactions in these two datasets. The statistics of the final processed datasets are shown in Table 1. We split 80% items of each basket as training data and the remaining 20% as test data for both of the datasets.

4.2 Experimental Settings

4.2.1 Evaluation metrics. We evaluate the performance of models by the Top K recommendation metrics, including the Recall@K, Precision@K, HR@K, and NDCG@K [18, 36]. We first compute the recommendation score for the given user u with all the items then full ranking is executed to generate the top K most possible items.

4.2.2 Baseline. We consider the following baselines for comparison:

Simple method:

- **PersonPop-k**: It is a basic method to return top k items from the training set in terms of the purchase frequency of a given user.

MF-based Methods:

- **BPR-MF** [34]: It is a method to model user and item interactions. The representation is learned by maximizing the distance between the user and its purchased and unpurchased items.
- **Triple2vec** [39]: It learns the user and item representation via the triplets (*item*, *item*, *user*) sampled for a single shopping basket.

NBR Methods:

Additionally, we modify the relevant NBR (Next-Basket Recommendation) methods to suit our within-basket recommendation setting. Specifically, we treat the current partially provided basket as the last basket in the shopping history for next-basket prediction.

---
[2]https://www.kaggle.com/c/instacart-market-basket-analysis
[3]https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset
[4]https://www.kaggle.com/c/acquire-valued-shoppers-challenge

- **Dream** [49]: It leverages recurrent neural networks to model the dynamics of users' behaviors and the sequential patterns between items.
- **TIFUKNN** [33]: It is a nearest neighbor-based model that outperforms deep recurrent neural networks in NBR. It relies on the similarity of the target user with other users and the purchase history of the target user.

GNN-based Methods:

- **LightGCN** [17]: It is a simplified model of NGCF [43]. Lightgcn directly uses the normalized summation of neighbors to perform aggregation on the graph, which greatly improves the recommendation performance.
- **BasConv** [29]: It constructs a UBI graph and then designs the heterogeneous aggregators on the graph to realize an informative message passing in representation learning.
- **MITGNN** [29]: It is the recent model focused on the within-basket recommendation task, which retrieves multiple intents across the defined basket graph to learn the representation of users and items.

Denoising Methods:

- **SGL** [49]: It combines the collaborative graph neural network filtering model with contrastive learning for recommendation by perturbing the graph structure through simple data augmentation operations on the graph structure.
- **CLEA** [33]: CLEA denoises the basket by automatically splitting the basket into positive and negative sub-baskets and using anchor-guided contrastive learning. We adopt the idea of CLEA and adapt the model to the within-basket recommendation task.

Note that we omit the comparison with the potential basket recommendation baseline PerNIR [2] as their objective focuses on predicting the next item for the current basket rather than complementing the entire basket.

4.2.3 Parameter settings. For fair comparisons, we adopt the following setting for all methods: the batch size is set to 1024; the embedding size is fixed to 128; all embedding parameters are initialized by Xavier initialization [13]; the hidden dimension is 64 for all methods. We optimize each baseline method according to the validation set. For LightGCN and SGL, we adopt 3 layers of propagation to achieve their best performance on all datasets. For BasConv model, 2 layers' aggregation is performed to have the best results. For our BNCL model, We fine-tune the hyperparameter r within the range of $[0, 0.1, 0.2, 0.5]$ and p within the range of $[0.1, 0.3, 0.5, 0.7]$. the coefficients for both the cross-behavior and the within-behavior contrastive learning are tuned in the range of $[1 \times 10^{-1}, 1 \times 10^{-3}, 1 \times 10^{-5}]$. Our model converges best when the learning rate is 5×10^{-4}, and the number of propagating layers is 2 for u-i and b-i graph. On Instacart, the coefficients for both cross-behavior and within-behavior contrastive learning are 0.1; On Tafeng, when the coefficients for cross-behavior and within-behavior contrastive learning are set to 1×10^{-2} and 1×10^{-3} respectively, the model reaches the best performance; On Valued-shoppers, we choose the coefficients as 1×10^{-4} and 1×10^{-5} respectively for cross-behavior and within-behavior contrastive learning.

4.3 Results (RQ1)

In this section, we compare the performance of several state-of-the-art baselines on the within-basket recommendation task on three real-world datasets and the results for $K = 40, 60$ are shown in Table 2. The baselines are arranged according to the different types of models.

We find that BNCL consistently outperforms other methods, which demonstrates its remarkable ability to extract informative representations by leveraging user-item interactions, basket-item interactions by effectively filtering out irrelevant or noisy information. The reported best-performing models are significant w.r.t. the second best performing with p-value < 0.05. What's more, the improvement of the baseline methods varies across different datasets (e.g. referring to the underlined results), while our method offers a general denoising approach that achieves stable enhancements.

The superiority of GNN-based models over classical models (BPRMF and Triple2vec) clearly demonstrates the significance of graph structure in modeling interactions and learning representations for within-basket recommendation tasks. However, we observed that the GNN-based method BasConv, which directly incorporates basket behavior, did not perform as well as expected. This observation suggests that poor-quality basket-item interactions can hinder representation learning and negatively impact performance.

Next basket recommendation methods TIFU-KNN, as well as CLEA, exhibited strong performance on several metrics as TIFU-KNN reaches the second-best results with recall and hit rate on ValuedShopper confirming the importance of the usage of the shopping history in basket recommendation. TIFU-KNN's poor performance on NDCG indicates that non-neural network-based methods tend to overlook the order of recommendations to some extent. Additionally, we find that NBR methods heavily rely on the number of purchased baskets so that it boosts better performance on ValuedShopper dataset where each user has more baskets on average.

It could be observed that methods such as SGL and CLEA which take denoising into account achieve better performance, showing the necessity of eliminating the effect of the noisy interaction in basket recommendation. Comparing SGL and CLEA, CLEA achieves better results since it focuses on denoising in the basket while SGL performs contrastive learning only on the user-item view, which proves the important role of basket denoising.

4.4 Ablation Study (RQ2)

In this section, we investigate the effectiveness of the proposed BNCL by evaluating the impact of different components. We denote the complementary basket hypergraph as B-I, the within-behavior contrastive learning with consistency-aware augmentation on the user-item interaction graph and basket view hypergraph as CA, and the cross-behavior contrastive learning as CL Fusion. We replace the consistency-aware augmentation with random edge perturbations and denote this model as BNCL$_{random}$. What's more, BNCL$_{add}$ represents the BNCL without hyperparameter r. Based on the typical user-item interaction graph, the checkmark under the corresponding module in Table 3 indicates whether this module was incorporated into the model. It can be seen that all the components are reasonably designed and essential to the final performance. We can observe that when any one of these components is removed, the
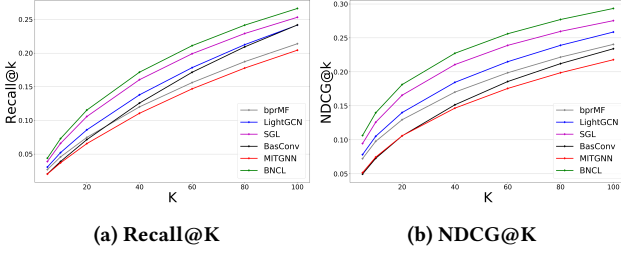
(a) Recall@K   (b) NDCG@K

Figure 3: Performance on two metrics of BNCL and a few most representative baselines w.r.t. K in the range of [5, 10, 20, 40, 60, 80, 100] on Instacart.

Table 3: Ablation Study of BNCL on Instacart.

B-I	CA	CL Fusion	Racall@60	Precision@60	HR@60	NDCG@60
-	-	-	17.83	2.33	70.31	21.50
✓	-	-	18.28	2.39	70.77	21.62
✓	-	✓	20.17	2.64	73.50	23.91
✓	✓	-	20.87	2.73	73.509	25.22
BNCL$_{random}$			20.56	2.70	72.48	25.29
BNCL$_{add}$			20.02	2.67	72.07	25.30
BNCL			**21.10**	**2.77**	**74.67**	**25.62**

performance drops accordingly on all metrics. The findings could be summarized as follows:

- Incorporating the basket hypergraph leads to a noticeable enhancement in performance, confirming that basket behavior offers additional valuable information for representation learning.
- Note that when integrating user purchase behavior and basket behavior, the utilization of cross-behavior contrastive learning yields the most significant improvement in recommendation performance compared to directly combining the embeddings learned from both behaviors (e.g., achieving a 10.59% increase in NDCG). This finding shows that cross-behavior contrastive learning contributes to reducing the effect of irrelevant information in the views of user purchase behavior and basket behavior by keeping the invariant and essential semantics of the items, which verifies the effectiveness of capturing the cooperative association between different views.
- The experimental results with random edge perturbations augmentation BNCL$_{random}$ and consistency-aware augmentation indicate that the consistency-aware augmentation will better identify the noise compared to a random structural augmentation and benefit the cross-behavior and within-behavior contrastive learning, which is consistent with our motivation in Section 3.4
- Compared with BNCL, the performance of BNCL$_{add}$ gets worse, which indicates that loosely adding the predictions of two separate views will include noise and ultimately results in suboptimal performance.

4.5 Case Study

4.5.1 Genralization Study (RQ3). We conduct experiments to verify the robustness and the generalization ability of the proposed

Table 4: Experimental comparisons of BNCL on Instacart with different backbones.

Backbone	Recall@60	Precision@60	HR@60	NDCG@60
BPRMF	15.61	2.03	66.05	19.88
BPRMF-BNCL	**16.72**	**2.51**	**67.69**	**20.39**
Fism	15.73	2.06	64.33	20.92
Fism-BNCL	**16.57**	**2.18**	**65.23**	**22.65**
LightGCN	17.83	2.33	70.31	21.50
LightGCN-BNCL	**21.10**	**2.77**	**74.67**	**25.62**

BNCL. First, we test BNCL with different K in the range of [5, 10, 20, 40, 60, 80, 100]. The results are presented in Figure 3, which shows that on all metrics, with the increasing value of K, our model consistently outperforms the baseline methods.

Second, we test our model with the different message-passing backbones on the user-item interaction graph. In the typical recommendation algorithms, user-item interaction data is often modeled as a user-item bipartite graph like what we employed in the user-item interaction graph part. Numerous algorithms have focused on representation learning using user-item graphs [6, 48]. Fism [21] proposed to improve the representation learning by training with the similarity matrix between items and for each training sample, it removes the direct link between the current user and its positive items when calculating the objective. We adopt Fism message-passing method and MF method on the user-item interaction modeling part of BNCL instead of the LightGCN message-passing to learn the embedding. The results on Istacart dataset are shown in Table 4, we can find that our model still performs the best even though the backbone has changed, which demonstrates the robustness and stability of our method.

4.5.2 Denoising Capability (RQ4). In this section, we further investigate the denoising performance of the proposed method. We introduce varying levels of noise to the user-item interaction graph and the basket hypergraph, specifically adding 20%, 40%, 60%, and 80% noisy interactions, and then observe the performance of BNCL. we consistently observe that BNCL outperforms both LightGCN and SGL across all noise ratios. As depicted in Figure 4, we note that as the ratio of added noise increases from 20% to 80%, our method experiences a mere 6.99% decline in Recall@60, whereas LightGCN's performance drops by 18.12% and SGL's by 9.00%. Similarly, the drop in NDCG@60 for our method is only 2.22%, whereas LightGCN experiences a 10.06% decline and SGL an 8.49% decline. These results serve as further evidence of the robustness of our model in handling noisy interactions. The evaluation under more advanced attack algorithms [10, 20] is left as future work.

5 RELATED WORK

In this section, we briefly review the related work on basket recommendation and contrastive learning for recommendation.

5.1 Basket Recommendation

Basket recommendation (BR) [12, 23] is to recommend a set of items that are mostly possible purchased by targeted users based on their
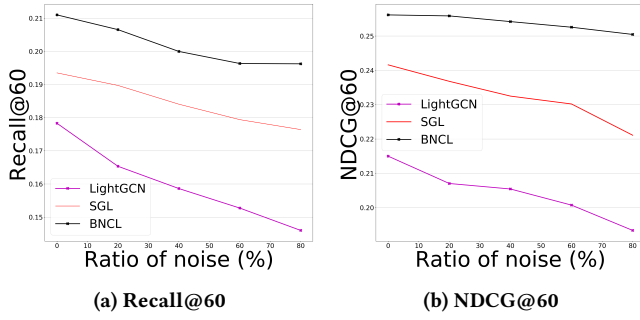
(a) Recall@60   (b) NDCG@60

Figure 4: Performance on two metrics of BNCL **and LightGCN with varying ratios of noise added on Instacart.**

shopping records. The basic idea is to capture correlations and perform the prediction through Collaborative Filtering (CF) methods [18, 46, 56] or Markov Chain (MC) methods [35, 41]. FPMC [35] is proposed to capture both sequential effects and long-term user taste where each user-specific transition is modeled by an underlying MC. Focus on predicting the items for the user's next baskets, DREAM [53] used an LSTM network [19] to capture the series features of the basket sequence. Some studies explore within-basket recommendations that also consider the content of the current basket. Triple2vec [39] improves the within-basket recommendation by constructing the training samples as triples of the user and the items. DBFM [26] contributes a basket recommendation solution based on factorization with a deep neural network. PerNIR [2] models the short-term interests of users represented by the current basket, as well as their long-term interests to address the task. Recently, GNN has shown its great potential to capture the interactions among the user and the item in representation learning [6, 37, 43]. Basconv [29] is proposed to capture heterogeneous interaction signals on a UBI graph by designing three different aggregators for user, basket and item entities. MITGNN [28] combines the translation-based model with the GNN to improve within-basket recommendation via retrieving the multi-intent pattern. In practice, noisy interactions are easy to occur during shopping behaviors and will hinder the capture of users' preferences and item properties while few works explicitly consider denoising in BR. For NBR, CLEA [33] used a denoising generator to denoise the baskets and then extract relevant items to enhance recommendation performance. However, the identification of noisy interactions and the elimination of their impact on heterogeneous behaviors in basket recommendation have been overlooked in the current literature.

5.2 Contrastive Learning for Recommendation

Contrastive learning aims to learn representation by minimizing the distance of positive instances while making negative instances far apart in the representation space [8], which has achieved great success on graph [15, 16, 52] and hypergraph [47] representation learning, as well as the application in recommender systems [49] and dense retrieval [22, 54, 55]. A Contrastive multi-view graph representation learning algorithm [16] is introduced for learning both node and graph-level representations by contrasting structural views of graphs. CLRec [57] bridged the theoretical gap between contrastive learning objective and traditional recommendation objective, which showed that directly performing contrastive learning

can help to reduce the exposure bias. Neighborhood-enriched Contrastive Learning (NCL) [27] explicitly incorporated the potential neighbors into contrastive pairs by introducing the neighbors of a user (or an item) from graph structure and semantic space respectively. CMP-PSP [50] effectively leveraged contrastive multi-view learning and pseudo-siamese networks to mitigate data sparsity and noisy interactions. CCFCRec [58] adopts contrastive collaborative filtering for cold-start item recommendation which applies contrastive learning to transfer the co-occurrence signals to the content CF module. KACL [40] performs contrastive learning across the user-item interaction view and KG view to include the knowledge graph in the recommendation while eliminating the noise it may introduce. Despite these advancements, the potential of contrastive learning in denoising within basket recommendation remains underexplored. Our work aims to harness the power of contrastive learning in behavior denoising and integrating diverse behaviors.

6 CONCLUSION

In this paper, we formulate the basket recommendation by integrating the basket behaviors into user-item interaction modeling with a light hypergraph message passing schema and a joint self-supervised learning paradigm. We systematically illustrate the noise issues in the basket recommendation problem. A general basket recommendation framework BNCL via noise-tolerated contrastive learning is proposed accordingly to improve robustness against the noise in BR. To be specific, we suppress the cross-behavior noise by making use of additional supervision signals with cross-behavior contrastive learning. Then to inhibit the within-behavior noise in the user and basket interactions, we propose to exploit invariant properties of the recommenders w.r.t augmentations through within-behavior contrastive learning. In addition, a novel consistency-aware augmentation approach is designed to better identify noisy interactions by comprehensively considering the two types of interactions. Extensive experimental results over three datasets on within-basket recommendation task show that our proposed method outperforms state-of-the-art baselines in terms of various ranking metrics.

One direct extension of our work is using BNCL as an effective tool to suppress the noise in multi-view learning and help the model fusion. Additionally, incorporating temporal dynamics and the order of the baskets into our current framework is a promising avenue for future research. Moreover, we'd like to test our method with more advanced backbones and self-supervised learning strategies. We're also interested in evaluating the bias [44] and fairness [45] of our approach under various settings.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17, 6 (2005), 734–749.

[2] Mozhdeh Ariannezhad, Ming Li, Sebastian Schelter, and Maarten de Rijke. 2023. A personalized neighborhood-based model for within-basket recommendation in grocery shopping. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 87–95.

[3] Ting Bai, Jian-Yun Nie, Wayne Xin Zhao, Yutao Zhu, Pan Du, and Ji-Rong Wen. 2018. An attribute-aware neural attentive model for next basket recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1201–1204.

[4] Marko Balabanović and Yoav Shoham. 1997. Fab: content-based, collaborative recommendation. *Commun. ACM* 40, 3 (1997), 66–72.

[5] Austin R Benson, Ravi Kumar, and Andrew Tomkins. 2016. Modeling user consumption sequences. In *Proceedings of the 25th International Conference on World Wide Web*. 519–529.

[6] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems* 33 (2020), 9912–9924.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[9] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 15750–15758.

[10] Minghong Fang, Guolei Yang, Neil Zhenqiang Gong, and Jia Liu. 2018. Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference*. 381–392.

[11] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3558–3565.

[12] Anna Gatzioura and Miquel Sànchez-Marrè. 2014. A case-based recommendation approach for market basket data. *IEEE Intelligent systems* 30, 1 (2014), 20–27.

[13] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[14] Riccardo Guidotti, Giulio Rossetti, Luca Pappalardo, Fosca Giannotti, and Dino Pedreschi. 2018. Personalized market basket prediction with temporal annotated recurring sequences. *IEEE Transactions on Knowledge and Data Engineering* 31, 11 (2018), 2151–2163.

[15] Yan Han, Yuning You, Wenqing Zheng, Scott Hoang, Tianxin Wei, Majdi Hassan, Tianlong Chen, Ying Ding, Yang Shen, and Zhangyang Wang. 2023. Graph Contrastive Learning: An Odyssey towards Generalizable, Scalable and Principled Representation Learning on Graphs. *Data Engineering* (2023), 78.

[16] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*. PMLR, 4116–4126.

[17] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[20] Mengdi Huai, Jianhui Sun, Renqin Cai, Liuyi Yao, and Aidong Zhang. 2020. Malicious attacks against deep reinforcement learning interpretations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 472–482.

[21] Santosh Kabbur, Xia Ning, and George Karypis. 2013. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 659–667.

[22] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:215737187

[23] Duc Trong Le, Hady W Lauw, and Yuan Fang. 2017. Basket-sensitive personalized item recommendation. IJCAI.

[24] Duc Trong Le, Hady W Lauw, and Yuan Fang. 2018. Modeling contemporaneous basket sequences with twin networks for next-item recommendation. IJCAI.

[25] Duc-Trong Le, Hady W Lauw, and Yuan Fang. 2019. Correlation-sensitive next-basket recommendation. (2019).

[26] Miao Li, Xuguang Bao, Liang Chang, and Tianlong Gu. 2022. Modeling personalized representation for within-basket recommendation based on deep learning. *Expert Systems with Applications* 192 (2022), 116383. https://doi.org/10.1016/j.eswa.2021.116383

[27] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*. 2320–2329.

[28] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and S Yu Philip. 2020. Basket recommendation with multi-intent translation graph neural network. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 728–737.

[29] Zhiwei Liu, Mengting Wan, Stephen Guo, Kannan Achan, and Philip S Yu. 2020. Basconv: Aggregating heterogeneous interactions for basket recommendation with graph convolutional neural network. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 64–72.

[30] Yunshan Ma, Yingzhi He, An Zhang, Xiang Wang, and Tat-Seng Chua. 2022. CrossCBR: Cross-view Contrastive Learning for Bundle Recommendation. *arXiv preprint arXiv:2206.00242* (2022).

[31] Mark Newman. 2010. *Networks: An Introduction*. Oxford University Press.

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[33] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The world is binary: Contrastive learning for denoising next basket recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 859–868.

[34] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[35] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[36] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.

[37] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[38] Jianhui Sun, Sanchit Sinha, and Aidong Zhang. 2023. Enhance Diffusion to Improve Robust Generalization. *arXiv preprint arXiv:2306.02618* (2023).

[39] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and recommending shopping baskets with complementarity, compatibility and loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1133–1142.

[40] Hao Wang, Yao Xu, Cheng Yang, Chuan Shi, Xin Li, Ning Guo, and Zhiyuan Liu. 2023. Knowledge-Adaptive Contrastive Learning for Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 535–543.

[41] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*. 403–412.

[42] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[43] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[44] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1791–1800.

[45] Tianxin Wei and Jingrui He. 2022. Comprehensive fair meta-learned recommender system. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1989–1999.

[46] Tianxin Wei, Ziwei Wu, Ruirui Li, Ziniu Hu, Fuli Feng, Xiangnan He, Yizhou Sun, and Wei Wang. 2020. Fast adaptation for cold-start collaborative filtering with meta-learning. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 661–670.

[47] Tianxin Wei, Yuning You, Tianlong Chen, Yang Shen, Jingrui He, and Zhangyang Wang. 2022. Augmentations in hypergraph contrastive learning: Fabricated and generative. *Advances in neural information processing systems* 35 (2022), 1909–1922.

[48] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*. 1437–1445.

[49] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[50] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Xiang Ao, Xin Chen, Xu Zhang, Fuzhen Zhuang, Leyu Lin, and Qing He. 2022. Multi-view multi-behavior contrastive learning in recommendation. In *Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part II*. Springer, 166–182.

[51] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1434–1443.

[52] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.

[53] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 729–732.

[54] Hansi Zeng, Surya Kallumadi, Zaid Alibadi, Rodrigo Nogueira, and Hamed Zamani. 2023. A Personalized Dense Retrieval Framework for Unified Information Access. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2023). https://api.semanticscholar.org/CorpusID:258332095

[55] Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum Learning for Dense Retrieval Distillation. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2022). https://api.semanticscholar.org/CorpusID:248426770

[56] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 11–20.

[57] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2021. Contrastive learning for debiased candidate generation in large-scale recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3985–3995.

[58] Zhihui Zhou, Lilin Zhang, and Ning Yang. 2023. Contrastive Collaborative Filtering for Cold-Start Item Recommendation. *arXiv preprint arXiv:2302.02151* (2023).

[59] Zhihua Zhu, Xinxin Fan, Xiaokai Chu, and Jingping Bi. 2020. Hgcn: A heterogeneous graph convolutional network-based deep learning model toward collective classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1161–1171.