Masked and Swapped Sequence Modeling for Next Novel Basket Recommendation in Grocery Shopping

Ming Li
University of Amsterdam
Amsterdam, The Netherlands
m.li@uva.nl

Mozhdeh Ariannezhad
AIRLab, University of Amsterdam
Amsterdam, The Netherlands
m.ariannezhad@uva.nl

Andrew Yates
University of Amsterdam
Amsterdam, The Netherlands
a.c.yates@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

ABSTRACT

Next basket recommendation (NBR) is the task of predicting the next set of items based on a sequence of already purchased baskets. It is a recommendation task that has been widely studied, especially in the context of grocery shopping. In next basket recommendation (NBR), it is useful to distinguish between repeat items, i.e., items that a user has consumed before, and explore items, i.e., items that a user has not consumed before. Most NBR work either ignores this distinction or focuses on repeat items. We formulate the *next novel basket recommendation* (NNBR) task, i.e., the task of recommending a basket that only consists of *novel items*, which is valuable for both real-world application and NBR evaluation. We evaluate how existing NBR methods perform on the NNBR task and find that, so far, limited progress has been made w.r.t. the NNBR task. To address the NNBR task, we propose a simple **b**i-directional **t**ransformer **b**asket **r**ecommendation model (BTBR), which is focused on directly modeling item-to-item correlations within and across baskets instead of learning complex basket representations. To properly train BTBR, we propose and investigate several masking strategies and training objectives: (i) item-level random masking, (ii) item-level select masking, (iii) basket-level all masking, (iv) basket-level explore masking, and (v) joint masking. In addition, an item-basket swapping strategy is proposed to enrich the item interactions within the same baskets. We conduct extensive experiments on three open datasets with various characteristics. The results demonstrate the effectiveness of BTBR and our masking and swapping strategies for the NNBR task. BTBR with a properly selected masking and swapping strategy can substantially improve NNBR performance.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Retrieval models and ranking*.

RecSys '23, September 18–22, 2023, Singapore, Singapore

ACM ISBN 979-8-4007-0241-9/23/09.
https://doi.org/10.1145/3604915.3608803

KEYWORDS

Next novel basket recommendation; Repetition and exploration

ACM Reference Format:
Ming Li, Mozhdeh Ariannezhad, Andrew Yates, and Maarten de Rijke. 2023. Masked and Swapped Sequence Modeling for Next Novel Basket Recommendation in Grocery Shopping. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3604915.3608803

1 INTRODUCTION

Next basket recommendation is a type of sequential recommendation that aims to recommend the next basket, i.e., set of items, to users given their historical basket sequences. Recommendation in a grocery shopping scenario is one of the main use cases of the NBR task, where users usually purchase a set of items instead of a single item to satisfy their diverse needs. Many methods, based on a broad range of underlying techniques (i.e., RNNs [3, 16, 21, 27, 43], self-attention [4, 32, 45], and denoising via contrastive learning [27]), have been proposed for, and achieve good performance on, the NBR task.

Next novel basket recommendation. A recent study [24] offers a new evaluation perspective on the next basket recommendation (NBR) task by distinguishing between *repetition* (i.e., recommending items that users have purchased before) and *exploration* (i.e., recommending items that are new to the user) tasks in NBR and points out the imbalance in difficulty between the two tasks. According to the analysis of existing methods in [24], the performance of many existing NBR methods mainly comes from being biased towards (i.e., giving more resources to) the repetition task and sacrificing the ability of exploration. Building on these insights, recent work on NBR has seen a specific focus on the pure repetition task [e.g., 20] as well the introduction of specific methods for the repetition task [1, 20].

Novelty and serendipity are two important objectives when evaluating recommendation performance [13, 18]. People might simply get tired of repurchasing the same set of items. Even when they engage in a considerable amount of repetition, there is still a large proportion of users who would like to try something new when shopping for groceries [24]. This phenomenon is especially noticeable for users with fewer transactions in their purchase history [1]. Therefore, one of the key roles of recommender systems is to assist

Table 1: Three types of basket recommendation.

Task	Target items	Recommended basket	Related work
NBR	Repeat items & novel items	Repeat items & novel items	[3, 4, 16, 21, 27, 27, 32, 43, 45]
NBRR	Repeat items	Only repeat items	[1, 20]
NNBR	Novel items	Only novel items	This paper

users in discovering potential novel items that align with their interests. However, in contrast to the pure repetition task, the pure exploration task in NBR remains under-explored. Besides, due to the difference in difficulty between the two tasks, many online e-commerce and grocery shopping platforms have started to design a "buy it again" service to isolate repeat items from the general recommendation.[1,2]

Motivated by the research gaps and real-world demands, we formulate the *next novel basket recommendation* (NNBR) task, which focuses on recommending a novel basket, i.e., a set of items that are new to the user, given the user's historical basket sequence. Different from the repetition task, which predicts the probability of repurchase from a relatively small set of items, the NNBR task needs to predict possible items from many thousands of candidates by modeling item-item correlations, which is more complex and difficult [24]. NNBR is especially relevant to the "Try Something New" concept in the grocery shopping scenario. Table 1 shows differences between three types of basket recommendation and positions our work.

From NBR to NNBR. The NNBR task can be seen as a sub-task of the conventional NBR task, in which NBR methods are designed to find all possible items (both *repeat items* and *novel items*) in the next basket. Therefore, it is possible to generate a novel basket by selecting only the top-k novel items predicted by NBR methods. To modify NBR methods for the NNBR task, an intuitive solution is to remove the repeat items from the ground-truth labels and train models only depending on the novel items in the ground-truth labels. Given this obvious strategy and given that many methods have already been proposed for NBR, an important question is: *If we already have an NBR model, do we need to train another model specifically for the NNBR task?* Surprisingly, we find that training specifically for exploration does not always lead to better performance in the NNBR task, and might even reduce performance in some cases.

BTBR: A bi-directional transformer basket recommendation method. In NNBR, item-to-item correlations are especially important, since we need to infer the utility of new items based on previously purchased items. Besides, a single basket is likely to address diverse needs of a user [37]. E.g., what a user would like to drink is more likely to depend on what he or she drank before rather than on the tooth paste they previously purchased. However, most existing NBR approaches [16, 21, 27, 43, 45] are two-stage methods, which first generate a basket-level representation [35], and then learn a temporal model based on basket-level representations, which will lead to information loss w.r.t. item-to-item correlations [21, 32, 45]. Some methods [21, 32, 45] learn partial item-to-item correlations based on the co-occurrence within the same or adjacent basket as auxiliary information beyond basket-level correlation learning. Instead of learning or exploiting complex basket representations, we learn item-to-item correlations from direct interactions among different items across different baskets. To do so, we propose a bi-directional transformer basket recommendation model (BTBR) that adopts a bi-directional transformer [36] and uses the shared basket position embedding to indicate items' temporal information.

Masking and training. To properly train BTBR, we propose and investigate several masking strategies and training objectives at different levels and tasks, as follows: (i) item-level random masking: a cloze-task loss [8, 34], in which we randomly mask the historical sequence at the item level; (ii) item-level select masking: a cloze-task loss designed for exploration, in which we first select the items we need to mask and then mask all the occurrences of the selected item; (iii) basket-level all masking: a general basket recommendation task loss, in which we mask and predict the complete last basket at the end of the historical sequence; (iv) basket-level explore masking: an explore-specific basket recommendation task loss, in which we remove the repeat items and only mask the novel items in the last basket of the historical sequence; and (v) joint masking: a loss that follows the pre-train-then-fine-tune paradigm, in which we first adopt item-level masking for the cloze task, then fine-tune the model using basket-level masking.

In addition, conventional sequential item recommendation usually assumes that the items in a sequence are strictly ordered and sequentially dependent. However, recent work [e.g., 5, 26, 39, 42] argues that the items may occur in any order, i.e., the order is flexible, and ignoring flexible orders might lead to information loss. Similarly, it is unclear whether the items that are being purchased across baskets have a strict order in the grocery shopping scenario. Thus, we propose an item swapping strategy that allows us to randomly move an item to another basket according to a certain ratio, which can enrich item interactions within the same basket.

We conduct extensive experiments on three publicly available grocery datasets to understand the effectiveness of the BTBR model and the proposed strategies on datasets with various repeat ratios and characteristics.

Our contributions. The main contributions of this paper are:

- To the best of our knowledge, we are the first to formulate and investigate the next novel basket recommendation (NNBR) task, which aims to recommend a set of novel items that meets a user's preferences in the next basket.

[1]After login, users may see a "buy it again" page on e-commerce platforms (see, e.g., Amazon https://amazon.com and grocery shopping platforms (see, e.g., Picnic https://picnic.app), where the platform collects repeat items. Similarly, in the grocery shopping scenario, "Try Something New" services also exist, where only novel items are recommended to the user.

[2]See, e.g., http://community.apg.org.uk/fileUploads/2007/Sainsburys.pdf for an example of the "Try Something New" concept in offline retail, and the Weekly New Recipe service at https://ah.nl/allerhande/wat-eten-we-vandaag/weekmenu for an example in online retail.

- We investigate the performance of several representative NBR methods w.r.t. the NNBR task and find (i) that training specifically for the exploration task does not always lead to better performance, and (ii) that limited progress has been made w.r.t. the NNBR task.
- We propose a simple bi-directional transformer basket recommendation (BTBR) model that learns item-to-item correlations across baskets.
- We propose several types of masking and item swapping strategies for optimizing BTBR for the NNBR task. Extensive experiments are done on three open grocery shopping datasets to assess the effectiveness of the proposed strategies. BTBR with a proper masking and swapping strategy is the new state-of-the-art method w.r.t. the NNBR task.

2 RELATED WORK

In this section, we describe two lines of research in the recommender systems literature that are related to our work: sequential recommendation and next basket recommendation.

Sequential recommendation. Sequential item recommendation has been widely studied for many years, and models [14, 15, 19, 23, 25, 31, 33, 41] with various deep learning techniques, e.g., RNN [14, 15], CNN [33], GNN [29, 41], contrastive learning [42], attention [23, 25] and self-attention [19, 31, 36] mechanism have been proposed. The self-attention (transformer) model [36] with multi-head attention shows strong performance in natural language processing, and SASRec [19] is the first sequential recommendation model that employs the self-attention mechanism. BERT4Rec [31] upgrades the left-to-right training scheme in SASRec and uses a bi-direction transformer with a Cloze task [34], which is the closest sequential recommendation method to this paper. Motivated by the success of BERT4Rec, some follow-up work has applied masked-item-prediction training to more specific scenarios [44].

However, BERT4Rec and follow-up work only focus on the item sequential recommendation with only random masking during training [44]. We extend BERT4Rec to the basket sequence setting and propose several types of masking strategies and training objectives that are specifically designed for the NNBR task. Furthermore, in this work we study the next novel basket recommendation task, where both historical interactions and the predicted target are baskets (sets of items). None of the sequential recommendation models listed above have been designed to handle a sequence of baskets.

Next basket recommendation. Next basket recommendation is a sequential recommendation task that addresses the sequence of baskets in the grocery shopping scenario. Existing methods can be classified into three types: frequency neighbor-based methods [9, 17], Markov chain (MC)-based methods [28], and deep learning-based methods [1, 3, 4, 16, 20–22, 27, 32, 37, 38, 40, 43, 45]. Recently, Li et al. [24] have evaluated and assessed NBR performance from a new repetition and exploration perspective; they find that the repetition task, i.e., recommending repeat items, is much easier than the exploration task, i.e., recommending explore items (a.k.a. novel items in this paper), besides the improvements of many recent methods come from the performance of the repetition task rather than better capturing correlations among items. Inspired by this finding, an NBR method [1] that only models the repetition behavior has been proposed, and an NBRR task [20] that only focuses on recommending repeat items has been formulated.

In this paper, we propose and formulate the next novel basket recommendation task that focuses on recommending novel items to the user, whereas all of the NBR methods mentioned above focus on the conventional NBR task, and their performance when generalized to the NNBR task remains unknown.

3 TASK FORMULATION

In this section, we describe and formalize the next novel basket recommendation task which is the focus of this paper.

Formally, given a set of users $U = \{u_1, u_2, \ldots, u_n\}$ and a set of items $I = \{i_1, i_2, \ldots, i_m\}$, $S_u = [B_u^1, B_u^2, \ldots, B_u^t]$ represents the historical interaction sequence for user u, where B_u^t represents a set of items $i \in I$ that user u purchased at time step t. For a user u, the *repeat item* i_u^{rep} is the item that user u has purchased before, which is defined as $i_u^{rep} \in I_u^{rep} = B_u^1 \cup B_u^2 \cup \ldots \cup B_u^t$, and the novel item i_u^{novel} is the item that user u have not purchased before, i.e., $i_u^{novel} \in I_u^{novel} = I - I_u^{rep}$.

The goal of the *next novel basket recommendation* task is to predict the following novel basket which only consists of novel items i_u^{novel} that the user would probably like, based on the user's past interactions S_u, that is,

$$P_u = \hat{B}_u^{t+1} = f(S_u) \tag{1}$$

where P_u denotes a recommended item list that *only consists of the novel items* i_u^{novel} *of user* u.

4 OUR METHOD

In this section, we first describe the base bi-directional transformer basket recommendation model (BTBR) we use, then introduce several types of masking strategies for the NNBR task, and finally describe an item swapping strategy.

4.1 Bi-directional transformer basket recommendation model

Learning basket representations [35] and modeling temporal dependencies across baskets are two key components in almost all neural-based NBR methods. Many NBR methods introduce complex architectures to learn representations for baskets in grocery shopping [4, 21, 27, 32, 45]. Instead of proposing more complex architectures to learn better basket representations and temporal dependencies, we want to simplify the model and only focus on the item-level correlations across different baskets, which helps us to infer novel items from users' historical items.

As a widely used method to model temporal dependencies, a recurrent neural network (RNN) [6, 10] requires passing information sequentially according to the temporal order, whereas there is no temporal order for items within the same basket, and basket-level representations at each timestamp are required [16, 21, 27, 43]. Another alternative method is the self-attention mechanism (a.k.a. transformer) [36], which is capable of learning the representations of every position by exchanging the information across all positions. Therefore, we adopt the bi-directional transformer [8, 36] as the backbone of our BTBR model, which not only allows us to learn item-to-item correlations from the direct interactions among items

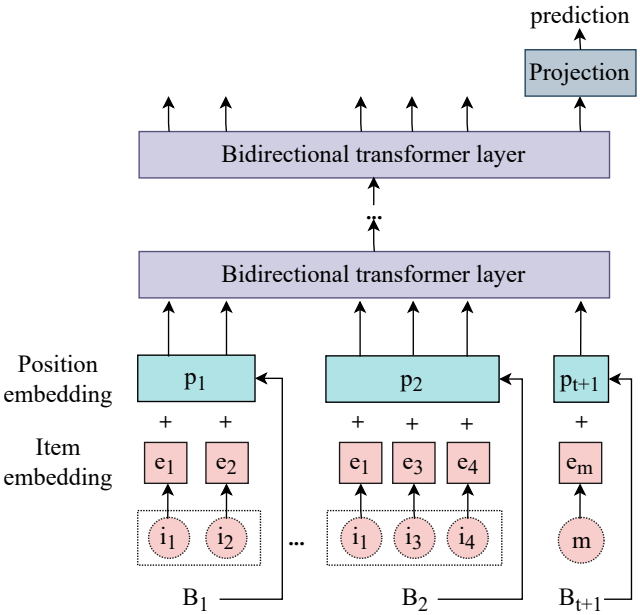


Figure 1: The overall architecture of the BTBR model.

across different baskets but is also able to handle basket sequence information in grocery shopping. The overall architecture of BTBR is shown in Figure 1.

Embedding layer. In order to use transformers [36] for NNBR, we first transfer a basket sequence to an item sequence via a "flatten" operation, e.g., $[\{i_1, i_2\}, \{i_1, i_3, i_4\}] \rightarrow [i_1, i_2, i_1, i_3, i_4]$. It has been shown that the positions of items are informative in the sequential recommendation scenario [19, 31]. Different from solutions in conventional item sequential recommendation, where each item is combined with its unique position embedding w.r.t. its position in the item sequence, we use a learnable position embedding for every basket, and items within the same basket will share the same position embedding. For example, given a basket sequence $S = [\{i_1, i_2\}, \{i_1, i_3, i_4\}, \{i_4, i_5\}]$, we first flatten S and get a sequence of item embeddings $E_i = [e_1^i, e_2^i, e_1^i, e_3^i, e_4^i, e_4^i, e_5^i]$, and a position embedding sequence $E_p = [e_1^p, e_2^p, e_3^p]$. Finally, the input sequence of transformer layer will be $E_{i,p} = [e_1^i + e_1^p, e_2^i + e_1^p, e_1^i + e_2^p, e_3^i + e_2^p, e_4^i + e_2^p, e_4^i + e_3^p, e_5^i + e_3^p]$. Note that the padding and truncating operations are also employed to handle sequences of various lengths.

Bi-directional transformer layer. The transformer architecture contains two sub-layers:

(1) *Multi-head attention layer*, which adopts the popular attention mechanism [36] and aggregates all items' embeddings across different baskets with adaptive weights.
(2) *Point-wise feed-forward layer*, which aims to endow nonlinearity and interactions between different latent dimensions.

We use stacked transformer layers to learn more complex item-to-item correlations, that is:

$$H^1 = \mathrm{Trm}(E_{i,p}), \ldots, H^L = \mathrm{Trm}(H^{L-1}), \tag{2}$$

where Trm denotes the bi-directional transformer layer, $H^L = [h_1^L, h_2^L, \ldots, h_d^L]$ denotes a representation sequence derived from the last transformer layer, and d denotes the maximum sequence length of input sequence $E_{i,p}$. Besides, residual connections [11], dropout [30], layer normalization [2], and GELU activation [12] are adopted to enhance the ability of representation learning. For more details about the bi-directional transformer layer, we refer to [19, 31, 36].

Prediction layer. After hierarchically exchanging information of all items across baskets using the transformer, we get $H^L \in \mathbb{R}^{m \times d}$, which contains the corresponding representations h^L for all items in the input sequence. Following [19, 31], we use the same item embedding $E_I \in \mathbb{R}^{m \times d}$ as the input layer to reduce the model size and alleviate the overfitting problem. For a masked position (item), we get its learned representation $h \in \mathbb{R}^d$ and compute the interaction probability distribution p of candidate items by:

$$p = \mathrm{Softmax}(hE^{\mathrm{T}} + b), \tag{3}$$

where E is the embedding matrix for candidate items and b denotes a bias term.

4.2 Masking strategy

Since there are repetition signals in the basket sequence, it is unclear whether these signals are merely noise/shortcuts or contain valuable information for the task of recommending novel items. After constructing the base model (BTBR), the challenging problem that needs to be addressed is how to properly train the model to improve its ability of finding novel items that meet users' interests. In this section, we propose four types of alternative masking strategies for the next novel basket recommendation task by considering different tasks and levels, as well as the repetition-exploration signals. Figure 2 shows examples of four types of masking strategies and Table 2 shows the characteristics of different training strategies.

Cloze task. The first type of training objective is a cloze task [34], i.e., "masked language model" in [8]. Specifically, we mask a proportion of items in the input sequence, i.e., replace each of them with a "mask token," and then try to predict the original items based on their contexts. We call this masking "item-level." Two main advantages of this item-level masking & training strategy are (i) it allows us to generate more item-level training samples by breaking the definition of "basket," and (ii) it learns both sides' information via the bi-directional transformer, which might allow the model to better capture item-to-item correlations. We first introduce two item-level masking strategies as follows:

(1) *Random*: This is a conventional masking strategy, which has been adopted in BERT4Rec [31]. Specifically, given a flattened item sequence, we randomly select several positions of the sequence and mask the corresponding items of the selected position according to mask ratio α as input.
(2) *Select*: One potential issue w.r.t. the above *Random* masking is that the masked items (prediction target) might still exist in the non-masked positions, so the model might mainly predict the masked item via its repetition information rather than inferring new items based on item-to-item correlations. Therefore, we propose the select masking strategy, which is specifically targeted at the exploration demand of the NNBR task. Specifically, given a flattened item sequence, we first derive the item set I in this sequence, then randomly select several items $i_m \in I$

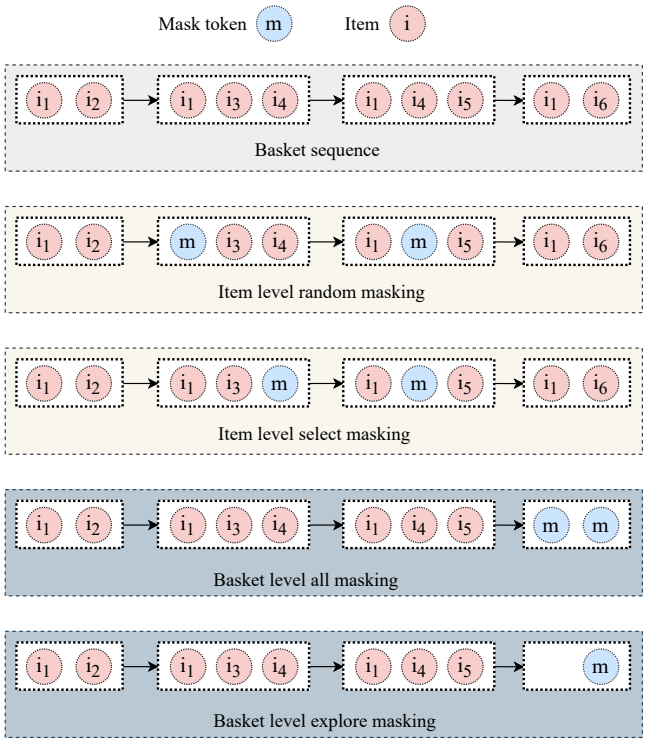


Figure 2: The original basket sequence (at the top) and four types of masking strategies.

according to mask ratio α, and finally mask all the occurrences of i_m in the sequence. Since there is no repetition information available, the model can only infer the targeted items, i.e., novel items, via learning the item-to-item correlations.

Basket recommendation task. Using the cloze task as the learning objective has limitations: (i) it is not able to fully respect the temporal dependencies of a sequence, since we can only use the historical information (left-side context) when we make the recommendation; and (ii) it is not specifically designed for the basket recommendation task and a mismatch might exist. Therefore, the second type of training objective we consider is the basket recommendation task, which masks the input sequence at the basket-level instead of item-level. Specifically, we mask the last basket and try to predict the items in this basket only based on the historical items (left-side information). Similarly, we propose another two basket-level masking strategies as follows:

(3) *All*: This masking strategy can be regarded as optimizing the model for the NBR task. Given a flattened item sequence, we find and mask all items, i.e., both novel items and repeat items in the last basket.
(4) *Explore*: This is a NNBR-specific masking strategy. Given a flattened item sequence, we find the items in the last basket, instead of masking all items, we only mask the novel items $i \in I^{novel}$ and remove the *repeat items* $i \in I^{rep}$. The model will be only optimized for finding all novel items in the future based on the historical basket sequence.

Table 2: Comparison of four types of masking strategies from three aspects, i.e., temporal orders, explore specific and amount of training signals.

Strategy	Strict temporal orders	Explore specific	Training signals ranking[3]
Item-Random	×	×	1
Item-Explore	×	✓	1
Basket-All	✓	×	2
Basket-Explore	✓	✓	3

Joint task. The pretrain-then-finetune paradigm has been widely adopted in NLP tasks. It is worth noting that item-level masking (the cloze task) and basket-level masking (the basket recommendation task) can also be combined as a joint masking strategy to employ the pretrain-then-finetune paradigm in NNBR, which first uses item-level masking strategy (i.e., self-supervised task) to get item correlations as the pre-train stage and then employ basket-level masking strategy (i.e., supervised task) to finetune it for the basket recommendation task.

Loss. Following [31], we select minimizing the negative log-likelihood loss as the training objective:

$$\mathcal{L} = \frac{1}{|I^m|} \sum_{i \in I^m} -\log p(i \mid S_u), \tag{4}$$

where I^m is the masked item set, $p(i \mid S_u, t)$ is the predicted probability of item i at position t.

Test and prediction. To predict a future basket (a set of items), we only need to add one masked token at the end of the user's item sequence, since items within the same basket share the same position embedding. In the NNBR task, the candidate items are novel items I^{new} that the user has not bought before, thus we use the embedding matrix w.r.t. the novel items of every user to compute the probabilities according to Eq. 3. Finally, we select top-K novel items with the highest scores as the recommendation list of the next novel basket.

4.3 Swapping strategy

In sequential recommendation, some work [5, 26, 39, 42] argues that the items in a sequence may not be sequentially dependent and different item orders may actually correspond to the same user intent. Ignoring flexible orders in sequential recommendation might lead to less accurate recommendations for scenarios where many items are not sequentially dependent [26, 39, 44]. In grocery shopping, the items purchased within the different baskets might not have rigid orders. To further understand if considering the flexible orders among items could further improve the performance w.r.t. the NNBR task, we propose the item swapping strategy to create augmentations for the BTBR.

Specifically, as illustrated in Figure 3, we randomly select items according to a swap ratio λ and then move them to another basket to enrich the items' interactions within the same basket. Besides, we introduce a hyper-parameter, i.e., swap hop γ, to control the

[3]As item-level masking can be seen as self-supervised learning, which is more flexible and can leverage more training signals than basket-level masking. Basket-explore has the least training signals as it can only use the novel items in the last basket.

basket distance of the swapping strategy. Note that we only perform the local swap strategy when using item-level masking (the cloze task) to train the model, since basket-level masking (the basket recommendation task) is designed to respect the sequential order and predict the future basket based on historical information.

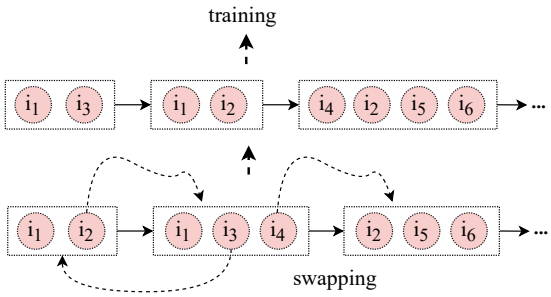


Figure 3: An example of the item swapping strategy.

5 EXPERIMENTS

5.1 Research questions

To understand the next novel basket recommendation task, and evaluate the performance of BTBR with different strategies, we conduct experiments to answer the following questions:

RQ1 How do existing NBR models perform w.r.t. the NNBR task? Does training specifically for the NNBR task lead to better performance?

RQ2 How does BTBR with different masking strategies perform compared to the state-of-the-art models?

RQ3 Does the swapping strategy contribute to the improvements?

RQ4 How do the hyper-parameters influence the models' performance and how different masking strategies affect the training dynamics?

RQ5 Is the joint masking strategy more robust than using the single masking strategy?

5.2 Experimental setup

Datasets. We evaluate the NNBR task on three publicly available grocery shopping datasets (TaFeng,[4] Dunnhumby,[5] and Instacart[6]), which vary in their repetition and exploration ratios. Following [24], we sample users whose basket length is between 3 and 50, and remove the least frequent items in each dataset. We also focus on the fixed size (10 or 20) next novel basket recommendation problem. In our experiments, we split the dataset across users, 80% for training, and 20% for testing, and leave 10% of the training users as the validation set. We repeat the splitting and experiments five times and report the average performance. The statistics of the processed datasets are shown in Table 3.

Baselines. We investigate the performance of six NBR baselines, which we select based on their performance on our chosen datasets in the analysis performed in [24]. Importantly, for a fair comparison, we do not include methods that leverage additional information [3, 4, 32].

[4]https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset
[5]https://www.dunnhumby.com/source-files/
[6]https://www.kaggle.com/c/instacart-market-basket-analysis/data

Table 3: Statistics of the processed datasets.

Dataset	#items	#users	Avg. basket size	Avg. #baskets per user	repeat ratio	explore ratio
TaFeng	11,997	13,858	6.27	6.58	0.188	0.812
Dunnhumby	3,920	22,530	7.45	9.53	0.409	0.591
Instacart	13,897	19,435	9.61	13.21	0.597	0.403

- **G-TopFreq:** A simple and effective method that recommends the top k most popular items in the dataset as the next basket for users.
- **TIFUKNN:** A state-of-the-art method that models the temporal dynamics of users' past baskets by using a KNN-based approach based on personalized frequency information (PIF) [17].
- **Dream:** A RNN-based method that gets basket representation using pooling strategy and employs RNN to model sequential behavior [43].
- **Beacon:** A RNN-based method that uses RNN to capture sequential behavior and uses correlation-sensitive basket encoder to consider intra-basket item correlations [21].
- **DNNTSP:** A state-of-the-art method that utilizes a graph neural network (GNN) and self-attention mechanisms to encode item-item relations across baskets and capture temporal dependencies [45].
- **CLEA:** A state-of-the-art method that uses contrastive learning and a GRU-based encoder to denoise and automatically extract items relevant to the target item [27].

Note that for the above baseline models (except G-TopFreq), we have two versions with different training methods, i.e., using all items in the last basket as training labels (Train-all), and only using novel items in the last basket as training labels (Train-explore).

Configurations. For the training-based baseline methods and TIFUKNN, we strictly follow the hyper-parameter setting and tuning strategy of their respective original papers. The embedding size is tuned on $\{16, 32, 64, 128\}$ for all training-based methods based on the validation set to achieve their best performance.

We use PyTorch to implement our model and train it using a TITAN X GPU with 12G memory. For BTBR, we set self-attention layers to 2 and their head number to 8, and tune the embedding size on $\{16, 32, 64, 128\}$. The Adam optimizer with a learning rate of 0.001 is used to update parameters. We set the batch size to 128 for the Tafeng and Dunnhumby datasets, and 64 for the Instacart dataset; we sweep the mask ratio α in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, local swap ratio in $\{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$ and swap hop γ in $\{1, 3, 5, 7, 9\}$.

Metrics. Two widely used metrics for the NBR problem are *Recall@k* and *nDCG@k*. In the NNBR task, *Recall* measures the ability to find all novel items that a user will purchase in the next basket; NDCG is a ranking metric that also considers the order of these novel items, i.e.,

$$\text{Recall@}K = \frac{1}{|U|} \sum_{u \in U} \frac{\left|P_u \cap T_u^{novel}\right|}{\left|T_u^{novel}\right|}, \tag{5}$$

$$\text{nDCG@}K = \frac{1}{|U|} \sum_{u \in U} \frac{\sum_{k=1}^{K} p_k / \log_2(k+1)}{\sum_{k=1}^{\min(K, |T_u^{novel}|)} 1/\log_2(k+1)}, \tag{6}$$

where U is a set of users who will purchase novel items in their next basket, T_u^{novel} is a set of ground-truth novel items of user u, p_k equals 1 if $P_u^k \in T_u^{novel}$, otherwise $p_k = 0$. P_u^k denotes the k-th item in the predicted basket P_u. Note that some methods might assign high scores w.r.t. the repeat items [24], to generate a novel basket, we fully remove the repeat items, then only rank and select top-k novel items as the recommended basket P_u to ensure a fair comparison, i.e., the recommended basket *only consists top-k novel items*.

5.3 Train-all and Train-explore (RQ1)

To answer RQ1, we employ two training strategies for each baseline method: (i) *Train-all*: we keep both repeat items and explore items as part of the ground-truth labels during training, which means that the model is trained to find all possible items in the next basket; and (ii) *Train-explore*: we remove the repeat items and only keep novel items in the ground-truth labels during training, which means the model is specifically trained to find novel items in the next basket. For the NNBR performance evaluation, we assess the models' ability to find novel items, which means the recommended novel basket consists of top-k novel items and there are no repeat items. We report the experimental results of different baseline methods in Table 4. We have three main findings.

First, we see that no method consistently outperforms all other methods across all datasets. On the Tafeng dataset, several NN-based methods (Dream-all, Dream-explore, Beacon-all, Beacon-explore, DNNTSP-all, CLEA-explore) fall in the top-tier methods group with quite good performance. On the Dunnhumby dataset, Beacon-explore achieves the best performance w.r.t. all metrics. On the Instacart dataset, TIFUKNN-explore is among the best-performing methods, which means that well-tuned neighbor-based models may outperform complex neural-based methods on some datasets w.r.t. the NNBR task [7, 24]. The performance of G-TopFreq is obviously the worst on the Tafeng and Dunnhumby dataset, however, its performance is quite competitive on the Instacart dataset, which indicates that the popularity information is very important w.r.t. the NNBR task in the scenario with a high repeat ratio.

Second, the improvements of recent methods achieved in NBR task do not always generalize to the NNBR task. Recent proposed methods (TIFUKNN, CLEA, DNNTSP) have surpassed the previous classic baselines (i.e., G-TopFreq, Dream, Beacon) by a large margin in conventional NBR task [17, 24, 27, 45], whereas, the improvements are relatively small or even missing on some datasets when handling the NNBR task. This indicates that the recently proposed methods make limited progress on finding novel items for the user and that their improvements mainly come from the repeat recommendation, which is consistent with the findings in [24].

Third, the NNBR performance changes diversely for different methods when changing from Train-all to Train-explore. Training and tuning existing NBR methods specifically for the NNBR task lead to significant or mild improvements in most cases, since the models do not need to deal with the repetition task and they are more targeted on finding novel items that meet users' preferences. Surprisingly, we find that DNNTSP-explore's performance is much worse than DNNTSP-all on the Tafeng and Dunnhumby datasets. We suspect that the underlying reason for this deterioration is that the repeat items (labels) contain useful item-to-item correlation signals that can be captured by the DNNTSP.[7] Since various NBR methods have distinct architectures, certain methods may gain more from tailored training for exploration, while others can grasp item-item correlations from repeat labels. Consequently, it is unwise to indiscriminately eliminate repeat labels during training.[8]

5.4 Effectiveness of BTBR (RQ2)

In this experiment, we evaluate the overall NNBR task performance of BTBR with different masking strategies, i.e., item-level random masking (item-random), item-level select masking (item-select), basket-level all masking (basket-all) and basket-level explore masking (basket-explore). The results of the comparison with the best baseline performances are shown in Table 5.[9] Based on the results, we have several observations. First, BTBR with the basket-all masking strategy (i.e., conventional next basket recommendation task) can significantly outperform the best baselines on the Tafeng and Instacart datasets, and achieve comparable performance on the Dunnhumby dataset. This result indicates that it may not be necessary to introduce basket representations, because only modeling item-to-item correlations is already effective for the NNBR task.

Second, there is no consistent best masking strategy across all datasets. On the Tafeng dataset, it is clear that basket-level masking outperforms item-level masking, where basket-all and basket-explore can respectively outperform and achieve the existing best performances w.r.t. each metric; however, using item-level masking leads to significant deterioration. On the Dunnhumby and Instacart datasets, BTBR with item-level masking strategies significantly outperforms the best performance achieved by baselines by a large margin, and is superior to BTBR with basket-level masking strategies. The above results show that the sequential order of items or baskets on the Tafeng dataset might be more strict than the order on the Dunnhumby and Instacart datasets, so using item-level masking, which fails to fully respect the sequential order and has poor performance on the Tafeng dataset.

Third, we can also observe that item-select masking achieves better performance than item-random masking w.r.t. all metrics across all datasets (paired t-test, $p < 0.05$), i.e., the improvements range from 4.1% to 9.0%, which demonstrates the effectiveness of our specifically designed item-select masking strategy for the NNBR task. In a sequence with many recurring items, the conventional random masking strategy could not ensure there is no masked item remaining in the other positions of the sequence, so the model might learn to predict the masked item based on the items' remaining occurrences, i.e., item self-relations. While the proposed item-select masking strategy will remove all occurrences of the same item, which can ensure that the masked items are novel items w.r.t. the remaining masked sequence, and the model has to infer the masked novel item via learning the masked item's relation with other items.

[7] Assume that one user's historical basket sequence is $[[a, b, c], [c, d], [a, c]]$, and next basket is $[b, e]$. Even though b is a repeat item, the model might be able to learn the correlation between b and other items in this historical sequence, which might help with the model's ability of finding novel items.

[8] This finding is important as it helps to avoid the potential issue of poor baselines. To ensure a fair comparison, NNBR practitioners should experiment with both strategies to train their baseline models and achieve best performances, instead of using an intuitive solution, i.e., removing repeat labels.

[9] To avoid confusion, we only mark the significant differences for comparison with the baselines in this table. More comparison results among different strategies can be found in the experimental analysis.

Table 4: Results of methods training for finding novel items, i.e., Train-explore, compared against the methods training for finding all items, i.e., Train-all. Boldface and underline indicate the best and the second best performing performance w.r.t. the NNBR task, respectively. Significant improvements and deteriorations of Train-explore over the corresponding Train-all baseline results are marked with $^{\uparrow}$ and $^{\downarrow}$, respectively. (paired t-test, $p < 0.05$).

Dataset	Metric	Train	G-Pop	TIFUKNN	Dream	Beacon	CLEA	DNNTSP
Tafeng	Recall@10	all	0.0587	0.0714	0.0960	0.0926	0.0870	**0.1024**
		explore	=	0.0911$^{\uparrow}$	<u>0.1021</u>$^{\uparrow}$	0.0967$^{\uparrow}$	0.1010$^{\uparrow}$	0.0940$^{\downarrow}$
	nDCG@10	all	0.0603	0.0662	0.0823	0.0789	0.0755	<u>0.0855</u>
		explore	=	0.0783$^{\uparrow}$	**0.0859**$^{\uparrow}$	0.0819$^{\uparrow}$	0.0857$^{\uparrow}$	0.0767$^{\downarrow}$
	Recall@20	all	0.0874	0.0926	0.1244	0.1252	0.1150	0.1245
		explore	=	0.1157$^{\uparrow}$	0.1244	**0.1257**	<u>0.1253</u>$^{\uparrow}$	0.1168$^{\downarrow}$
	nDCG@20	all	0.0703	0.0738	0.0928	0.0909	0.0861	<u>0.0943</u>
		explore	=	0.0876$^{\uparrow}$	0.0939	0.0929	**0.0952**$^{\uparrow}$	0.0858$^{\downarrow}$
Dunnhumby	Recall@10	all	0.0468	0.0497	0.0494	0.0499	0.0499	0.0514
		explore	=	0.0498	0.0506	**0.0529**$^{\uparrow}$	<u>0.0520</u>$^{\uparrow}$	0.0472$^{\downarrow}$
	nDCG@10	all	0.0397	0.0409	0.0409	0.0411	0.0376	<u>0.0415</u>
		explore	=	0.0411	0.0385	**0.0428**$^{\uparrow}$	0.0404$^{\uparrow}$	0.0378$^{\downarrow}$
	Recall@20	all	0.0701	0.0745	0.0744	0.0804	0.0711	0.0782
		explore	=	0.0746	0.0791	**0.0813**	<u>0.0807</u>$^{\uparrow}$	0.0739
	nDCG@20	all	0.0491	0.0505	0.0505	<u>0.0532</u>	0.0479	0.0524
		explore	=	0.0506	0.0502	**0.0546**	0.0521$^{\uparrow}$	0.0484$^{\downarrow}$
Instacart	Recall@10	all	0.0430	0.0425	0.0440	0.0454	0.0394	0.0414
		explore	=	**0.0494**$^{\uparrow}$	0.0455	0.0460	<u>0.0469</u>$^{\uparrow}$	0.0419
	nDCG@10	all	0.0359	0.0346	0.0356	0.0388	0.0302	0.0335
		explore	=	**0.0400**$^{\uparrow}$	0.0355	<u>0.0387</u>	0.0369$^{\uparrow}$	0.0341
	Recall@20	all	0.0685	0.0649	0.0690	0.0733	0.0626	0.0635
		explore	=	<u>0.0755</u>$^{\uparrow}$	0.0719	0.0741	**0.0764**$^{\uparrow}$	0.0642
	nDCG@20	all	0.0455	0.0431	0.0452	0.0499	0.0394	0.0424
		explore	=	<u>0.0500</u>$^{\uparrow}$	0.0462	**0.0501**	0.0484$^{\uparrow}$	0.0431

Finally, it can also be seen that basket-explore masking, which is specifically targeted at the NNBR task, does not lead to any improvements on the Tafeng and Dunnhumby datasets, and results in a decrease in performance on the Instacart dataset, compared with basket-all masking. This result again verifies the findings in Section 5.3 and indicates that masking and training BTBR specifically for the NNBR task may be suboptimal, since the repeat item labels may also be helpful with item-to-item correlations modeling.

5.5 Effectiveness of the item swapping strategy (RQ3)

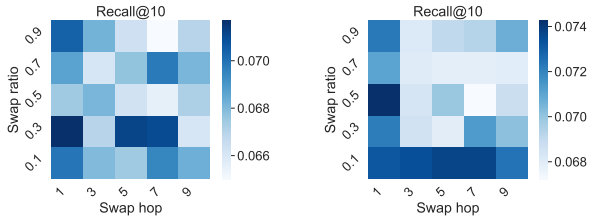
In this section, we conduct experiments to verify the effectiveness of the swapping strategy, and the results are shown in Table 5. We find that adding a swapping strategy on top of item-random and item-select leads to a decrease in performance on the Tafeng dataset. At the same time, we note that adding a swapping strategy on top of item-random and item-select leads to better performance on the Dunnhumby and Instacart datasets (paired t-test, $p < 0.05$). These results are not surprising, since the swapping strategy will not only enrich the item interactions within the basket, but also has a risk of introducing noise w.r.t. the temporal information. The sequential order is relatively strict on the TaFeng dataset (see Section 5.4), and the model can not benefit from the swap strategy.

We further investigate the influence of hyper-parameters of the swapping strategy, i.e., swap ratio and swap hop. Figure 4 shows a heatmap w.r.t. Recall@10 on different datasets when swap ratio ranges within $[0.1, 0.3, 0.5, 0.7, 0.9]$ and swap hop ranges within $[1, 3, 5, 7, 9]$. We observe that training with both high swap ratio and swap hop (the upper-right of the heatmap) leads to poor performance on the Tafeng and Dunnhumby dataset. When it comes to the Instacart dataset, better performance is achieved via using a high swap-hop. The repeat ratio on Instacart dataset is high, which means that the user's interest is relatively stable and swapping across adjacent baskets will not help, so a higher swap hop is preferred to enrich item interactions within the basket on this dataset.

Given the above findings, there is a trade-off between enriching the item interactions within baskets and respecting the original temporal order information, so it is reasonable to search for the optimal swap hyper-parameters to get the highest performance on different datasets in practice.

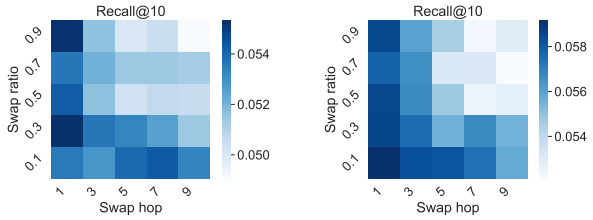
Table 5: Results of BTBR method with different masking strategies compared against the best performance of baseline method training for each metric w.r.t. NNBR task. Boldface and underline indicate the best and the second best performing performance w.r.t. the NNBR task, respectively. Significant improvements and deteriorations of over the best baseline results are marked with $^{\uparrow}$ and $^{\downarrow}$, respectively (paired t-test, $p < 0.05$). ▲% shows the improvements against the best performing baseline.

Dataset	Metric	Best	Item level				Basket level		Joint
			Random	Select	Random swap	Select swap	All	Explore	Pretrain-Finetune
Tafeng	Recall@10	0.1024	0.0736$^{\downarrow}$	0.0801$^{\downarrow}$	0.0717$^{\downarrow}$	0.0746$^{\downarrow}$	<u>0.1056</u>$^{\uparrow}$	0.1032	**0.1057**$^{\uparrow}$**(3.2%)**
	nDCG@10	0.0859	0.0597$^{\downarrow}$	0.0651$^{\downarrow}$	0.0587$^{\downarrow}$	0.0605$^{\downarrow}$	<u>0.0869</u>	0.0860	**0.0870 (1.3%)**
	Recall@20	0.1257	0.0977$^{\downarrow}$	0.1036$^{\downarrow}$	0.0895$^{\downarrow}$	0.0911$^{\downarrow}$	<u>0.1292</u>$^{\uparrow}$	0.1271	**0.1353**$^{\uparrow}$**(7.6%)**
	nDCG@20	0.0952	0.0691$^{\downarrow}$	0.0739$^{\downarrow}$	0.0685$^{\downarrow}$	0.0688$^{\downarrow}$	<u>0.0970</u>$^{\uparrow}$	0.0957	**0.0973**$^{\uparrow}$**(2.2%)**
Dunnhumby	Recall@10	0.0529	0.0548$^{\uparrow}$	0.0572$^{\uparrow}$	0.0553$^{\uparrow}$	<u>0.0592</u>$^{\uparrow}$	0.0524	0.0521	**0.0593**$^{\uparrow}$**(12.1%)**
	nDCG@10	0.0428	0.0439$^{\uparrow}$	0.0461$^{\uparrow}$	0.0443$^{\uparrow}$	**0.0469**$^{\uparrow}$	0.0427	0.0424	<u>0.0468</u>$^{\uparrow}$(9.3%)
	Recall@20	0.0813	0.0847$^{\uparrow}$	0.0891$^{\uparrow}$	0.0867$^{\uparrow}$	**0.0924**$^{\uparrow}$	0.0815	0.0806	<u>0.0915</u>$^{\uparrow}$(12.5%)
	nDCG@20	0.0546	0.0560$^{\uparrow}$	0.0587$^{\uparrow}$	0.0571$^{\uparrow}$	**0.0598**$^{\uparrow}$	0.0540	0.0532	<u>0.0596</u>$^{\uparrow}$(9.2%)
Instacart	Recall@10	0.0494	0.0554$^{\uparrow}$	0.0583$^{\uparrow}$	0.0572$^{\uparrow}$	**0.0600**$^{\uparrow}$	0.0539$^{\uparrow}$	0.0455$^{\downarrow}$	<u>0.0598</u>$^{\uparrow}$(21.1%)
	nDCG@10	0.0400	0.0445$^{\uparrow}$	0.0474$^{\uparrow}$	0.0458$^{\uparrow}$	**0.0486**$^{\uparrow}$	0.0426$^{\uparrow}$	0.0387$^{\downarrow}$	<u>0.0478</u>$^{\uparrow}$(19.5%)
	Recall@20	0.0764	0.0887$^{\uparrow}$	0.0924$^{\uparrow}$	0.0898$^{\uparrow}$	**0.0935**$^{\uparrow}$	0.0846$^{\uparrow}$	0.0734$^{\downarrow}$	<u>0.0934</u>$^{\uparrow}$(22.3%)
	nDCG@20	0.0501	0.0573$^{\uparrow}$	0.0607$^{\uparrow}$	0.0583$^{\uparrow}$	**0.0616**$^{\uparrow}$	0.0551$^{\uparrow}$	0.0485$^{\downarrow}$	<u>0.0613</u>$^{\uparrow}$(22.4%)



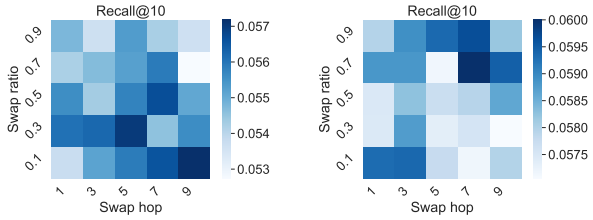
(a) Tafeng (item-random)

(b) Tafeng (item-select)



(c) Dunnhumby (item-random)

(d) Dunnhumby (item-select)



(e) Instacart (item-random)

(f) Instacart (item-select)

Figure 4: Performance heatmap with different swap hops and swap ratios.

5.6 Effect of mask ratio and training dynamics (RQ4)

We investigate the effect of mask ratio and analyze how the performance changes as training goes on to further understand the properties of different masking strategies.

Mask ratio. The mask ratio α when using item-level masking is a hyper-parameter that is worth discussing. Figure 5 shows the Recall@10 when the mask ratio ranges within $[0.1, 0.3, 0.5, 0.7, 0.9]$. We can observe that item-select outperforms item-random with the same mask ratio in most cases. We also see that the optimal mask ratio is 0.1 for item-random and item-select, and the optimal mask ratio is much higher (0.5, 0.7) on the Dunnhumby and Instacart datasets. We suspect that a higher mask ratio is preferred in the NNBR task when the dataset has long interaction records for the users.

Training dynamics. Figure 6 shows how the Recall@10 evolves as training goes when using different masking strategies. First, it is obvious that basket-level masking achieves its best performances very fast, and then drops much earlier than item-level masking. This is because the training labels of basket-level masking are static, which can easily lead to overfitting, while the training labels of item-level masking are dynamic, which alleviates overfitting. Second, compared to basket-all masking, basket-explore masking further aggravates the overfitting problem via removing the repeat items (labels), which might lead to a performance decrease, especially in the scenario with a high repeat ratio. Finally, the performance of item-random and item-select evolves similarly on the Tafeng dataset, since the repeat ratio on it is small. On the Dunnhumby and Instacart datasets, item-random masking results in overfitting earlier than the item-select masking, since the masked item might still exist in other positions of the masked sequence and the model will rely more on the repeat item prediction instead of inferring novel items, as the repetition prediction task is relatively easier [24].

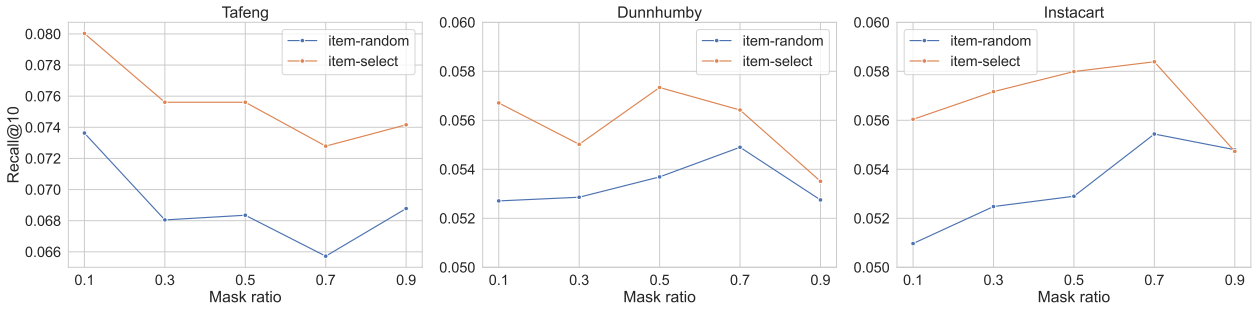


Figure 5: Performance of BTBR with item-random strategy and item-select masking strategy with various mask ratios.

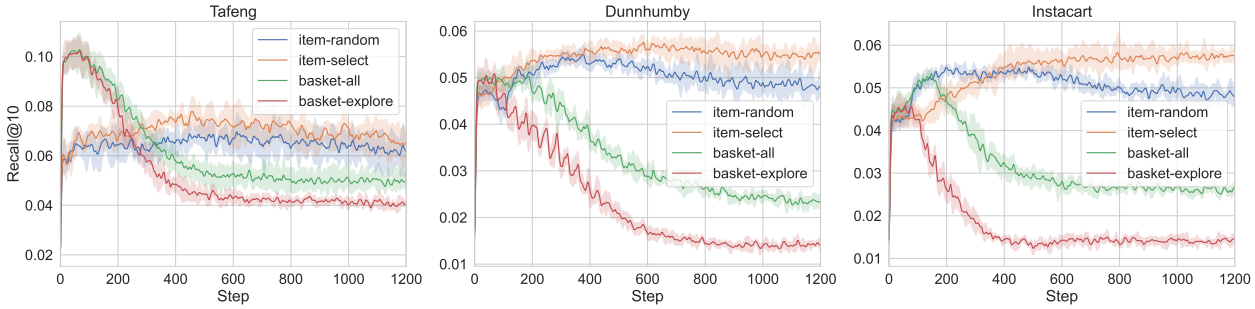


Figure 6: The training progress w.r.t. Recall@10 of BTBR with different masking strategies on three datasets.

5.7 Effectiveness of joint masking (RQ5)

So far, we have built a comprehensive understanding of different masking strategies and realize that no single masking strategy is optimal in all cases, due to the diverse characteristics of datasets. Now, we conduct experiments to evaluate the effectiveness of joint masking (training), i.e., pre-train the model using item-select masking, then fine-tune the model using basket-all masking. The results are also shown in Table 5. We find that BTBR with joint masking consistently outperforms the best performance obtained by existing baselines across datasets; the improvements range from 1.3% to 7.6% on Tafeng dataset, from 9.2% to 12.5% on Dunnhumby dataset and from 19.5% to 22.4% on Instacart dataset. Joint masking does not lead to further improvements compared with a single optimal strategy, i.e., basket-all on the Tafeng dataset and item-select with swap on the Dunnhumby and Instacart datasets, in most cases.[10] The joint masking strategy under the pretrain-then-finetune paradigm is still valuable due to its robustness w.r.t. NNBR task (i.e., it consistently achieves the best performance) on various datasets with different characteristics.

6 CONCLUSION

We have formulated the next novel basket recommendation task, i.e., the task of recommending novel items to users given historical interactions. The task has practical applications, and helps us to evaluate an NBR model's ability to find novel items for a given user. To understand the performance of existing NBR methods on the NNBR task, we have evaluated several NBR models with two training methods, i.e., Train-all and Train-explore. To address the NNBR task, we have proposed a bi-directional transformer basket recommendation model (BTBR), which uses a bi-directional transformer to directly model item-to-item correlations across different baskets. To train BTBR, we have designed five types of masking strategies and training objectives considering different levels: (i) item-level random masking, (ii) item-level select masking, (iii) basket-level all masking, (iv) basket-level explore masking, and (v) joint masking.

To further improve the BTBR performance, we also proposed an item swapping strategy to enriching item interactions.

We have conducted extensive experiments on three datasets. Concerning existing NBR methods we have found that: (i) the performance on the NNBR task differs widely between existing NBR methods; (ii) the performance of existing methods on the NNBR task leaves considerable room for improvement, and the top performing methods on the NNBR task are different from the top performers on the NBR task; and (iii) training specifically for the NNBR task by removing repeat items from the ground truth labels does not lead to consistent improvements in performance.

Concerning our newly proposed BTBR method, we have found that: (i) BTBR with a properly selected masking and swapping strategy can substantially improve the NNBR performance; (ii) there is no consistent best masking level for BTBR across all datasets; (iii) the proposed item-select masking strategy outperforms the conventional item-random masking strategy on the NNBR task; the item-basket swapping strategy can further improve NNBR performance; and (iv) a joint masking strategy is robust on various datasets but does not lead to further improvements compared to a single level masking strategy.

[10]The highest and second-highest scores in Table 5 are essentially at the same level and there is no significant difference between the joint training strategy and the single optimal strategy on each dataset in terms of performance.

A broader implication of our work is that blindly training specifically for the proposed recommendation task might lead to sub-optimal performance and it is necessary to consider various training objectives on diverse datasets. Another implication is that it is important to consider the differences between repetition behavior and exploration behavior when designing recommendation models for the grocery shopping scenario.

One limitation of this paper is that we only focus on the grocery shopping scenario. An obvious avenue for future work, therefore, is to extend the proposed item-select masking strategy to sequential item recommendation scenarios, and investigate if it can outperform the widely used item-random masking strategy w.r.t. finding novel items.

REPRODUCIBILITY

We share both our processed dataset and the source code used to produce the results in this paper at https://github.com/liming-7/Mask-Swap-NNBR.

ACKNOWLEDGMENTS

This research was (partially) funded by the China Scholarship Council (grant #20190607154), the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research (NWO), https://hybrid-intelligence-centre.nl, and project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Mozhdeh Ariannezhad, Sami Jullien, Ming Li, Min Fang, Sebastian Schelter, and Maarten de Rijke. 2022. ReCANet: A Repeat Consumption-Aware Neural Network for Next Basket Recommendation in Grocery Shopping. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1240–1250.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *arXiv preprint arXiv:1607.06450* (2016).

[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 and Development in Information Retrieval*. 1201–1204.

[4] Yongjun Chen, Jia Li, Chenghao Liu, Chenxi Li, Markus Anderle, Julian McAuley, and Caiming Xiong. 2021. Modeling Dynamic Attributes for Next Basket Recommendation. *arXiv preprint arXiv:2109.11654* (2021).

[5] Mingyue Cheng, Fajie Yuan, Qi Liu, Xin Xin, and Enhong Chen. 2021. Learning Transferable User Representations with Sequential Behaviors via Contrastive Pre-training. In *2021 IEEE International Conference on Data Mining (ICDM)*. 51–60.

[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-decoder Approaches. *arXiv preprint arXiv:1409.1259* (2014).

[7] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 101–109.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Guglielmo Faggioli, Mirko Polato, and Fabio Aiolli. 2020. Recency Aware Collaborative Filtering for Next Basket Recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 80–87.

[10] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural computation* 12, 10 (2000), 2451–2471.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[12] Dan Hendrycks and Kevin Gimpel. 2016. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *arXiv preprint arXiv:1606.08415* (2016).

[13] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[14] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 843–852.

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939* (2015).

[16] Haoji Hu and Xiangnan He. 2019. Sets2Sets: Learning from Sequential Sets with Neural Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1491–1499.

[17] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling Personalized Item Frequency Information for Next-basket Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1071–1080.

[18] Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-accuracy Objectives in Recommender Systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 1 (2016), 1–42.

[19] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. 197–206.

[20] Ori Katz, Oren Barkan, Noam Koenigstein, and Nir Zabari. 2022. Learning to Ride a Buy-Cycle: A Hyper-Convolutional Model for Next Basket Repurchase Recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 316–326.

[21] Duc-Trong Le, Hady W. Lauw, and Yuan Fang. 2019. Correlation-sensitive Next-basket Recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2808–2814.

[22] Youfang Leng, Li Yu, Jie Xiong, and Guanyu Xu. 2020. Recurrent Convolution Basket Map for Diversity Next-Basket Recommendation. In *International Conference on Database Systems for Advanced Applications*. 638–653.

[23] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[24] Ming Li, Sami Jullien, Mozhdeh Ariannezhad, and Maarten de Rijke. 2023. A Next Basket Recommendation Reality Check. *ACM Transactions on Information Systems* 41, 4 (October 2023), Article 116.

[25] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1831–1839.

[26] Mingda Qian, Xiaoyan Gu, Lingyang Chu, Feifei Dai, Haihui Fan, and Bo Li. 2022. Flexible Order Aware Sequential Recommendation. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*. 109–117.

[27] 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.

[28] 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.

[29] 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.

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[31] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1441–1450.

[32] Leilei Sun, Yansong Bai, Bowen Du, Chuanren Liu, Hui Xiong, and Weifeng Lv. 2020. Dual Sequential Network for Temporal Sets Prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1439–1448.

[33] Jiaxi Tang and Ke Wang. 2018. Personalized Top-n Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 565–573.

[34] Wilson L. Taylor. 1953. "Cloze Procedure": A New Tool for Measuring Readability. *Journalism Quarterly* 30, 4 (1953), 415–433.
[35] Vojtěch Vančura. 2021. Neural Basket Embedding for Sequential Recommendation. In *Fifteenth ACM Conference on Recommender Systems*. 878–883.
[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. *arXiv preprint arXiv:1706.03762* (2017).
[37] 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.
[38] Pengfei Wang, Yongfeng Zhang, Shuzi Niu, and Jiafeng Guo. 2019. Modeling Temporal Dynamics of Users' Purchase Behaviors for Next Basket Prediction. *Journal of Computer Science and Technology* 34, 6 (2019), 1230–1240.
[39] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. *arXiv preprint arXiv:2001.04830* (2019).
[40] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2020. Intention Nets: Psychology-inspired User Choice behavior Modeling for Next-basket Prediction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. 6259–6266.
[41] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based Recommendation with Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.
[42] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1259–1273.
[43] 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.
[44] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2022. Self-Supervised Learning for Recommender Systems: A Survey. *arXiv preprint arXiv:2203.15876* (2022).
[45] Le Yu, Leilei Sun, Bowen Du, Chuanren Liu, Hui Xiong, and Weifeng Lv. 2020. Predicting Temporal Sets with Deep Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1083–1091.