The World is Binary: Contrastive Learning for Denoising Next Basket Recommendation

Yuqi Qin
Beijing University of Posts and Telecommunications
Beijing, China
yuqiqin@bupt.edu.cn

Pengfei Wang*
Beijing University of Posts and Telecommunications
Beijing, China
wangpengfei@bupt.edu.cn

Chenliang Li*
School of Cyber Science and Engineering, Wuhan University
Wuhan, China
cllee@whu.edu.cn

ABSTRACT

Next basket recommendation aims to infer a set of items that a user will purchase at the next visit by considering a sequence of baskets he/she has purchased previously. This task has drawn increasing attention from both the academic and industrial communities. The existing solutions mainly focus on sequential modeling over their historical interactions. However, due to the diversity and randomness of users' behaviors, not all these baskets are relevant to help identify the user's next move. It is necessary to denoise the baskets and extract credibly relevant items to enhance recommendation performance. Unfortunately, this dimension is usually overlooked in the current literature.

To this end, in this paper, we propose a **C**ontrastive **Lea**rning Model (named **CLEA**) to automatically extract items relevant to the target item for next basket recommendation. Specifically, empowered by Gumbel Softmax, we devise a denoising generator to adaptively identify whether each item in a historical basket is relevant to the target item or not. With this process, we can obtain a *positive* sub-basket and a *negative* sub-basket for each basket over each user. Then, we derive the representation of each sub-basket based on its constituent items through a GRU-based context encoder, which expresses either relevant preference or irrelevant noises regarding the target item. After that, a novel two-stage anchor-guided contrastive learning process is then designed to simultaneously guide this relevance learning without requiring any item-level relevance supervision. To the best of our knowledge, this is the first work of performing item-level denoising for a basket in an end-to-end fashion for next basket recommendation. Extensive experiments are conducted over four real-world datasets with diverse characteristics. The results demonstrate that our proposed CLEA achieves significantly better recommendation performance than the existing state-of-the-art alternatives. Moreover, further analysis also shows that CLEA can successfully discover the real relevant items towards the recommendation decision.

*Corresponding author.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada.

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462836

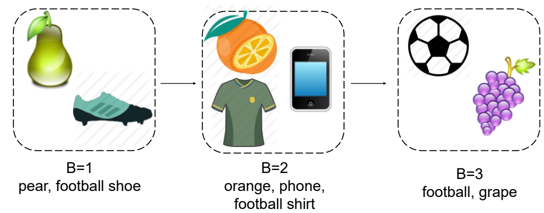

Figure 1: A motivating example to illustrate the necessary of making data denoising for next basket recommendation. Given the previous two baskets (B=1 and B=2), we aim to recommend items that user will buy in the next basket (i.e., B=3).

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Next basket recommendation; Contrastive Learning; Item-level denoising

ACM Reference Format:
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 (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462836

1 INTRODUCTION

Next basket recommendation is a fundamental task for E-Commerce platforms [13]. Generally, given a sequence of historical baskets for a user, where each basket contains a collection of items, we would like to recommend a shopping list that the user may purchase next. In this specific scenario, intra-basket and inter-basket item interactions are important factors need to be considered, as items a user purchase in the future may usually depend strongly on the items he/she has interacted with in the past. However, due to the diversity and randomness of user behaviors, the baskets often contain many items that are irrelevant to his/her next choice, which tends to overwhelm the useful signals from a few truly relevant ones. Therefore, a good recommender should be able to extract these items relevant to the target item to infer a correct recommendation [27].

Though promising, we argue that it is non-trivial to perform denoising due to the following two aspects: 1) Since there is no direct mechanism to collect item-level relevance information in basket recommendation scenarios, it is difficult to train an effective denoising model to enhance user understanding without recruiting any item-level supervision signal. 2) The denoising process should be context-aware such that the relevance of a historical item should be dependent on the target one. In addition, the randomness of user behaviors may also bring some items irrelevant to the task, which certainly aggravates the relevance learning process.

To explain this, we give an illustrative example in Figure 1. Given the previous two baskets (B=1 and B=2), we aim to recommend items that this user will buy in the next basket (i.e., B=3). It is interesting to see that item *phone* in the second basket should be neglected as it has no obvious relation with all items in the target basket. In addition, though *football shoe* and *football shirt* indicate strong connections with *football*, it turns to distract the preference learning towards item *grape*.

To distill the credible item relevance for a good recommendation, many research efforts have been devoted to sequential pattern mining technique [6]. Despite effectiveness, these techniques usually lack context-aware modeling and suffer from producing optimal results concerning the trade-offs between support and confidence [28]. Recently, with the prosperity of deep neural network, various approaches are adopted to composite the relevant signals from the historical interactions for recommendation [16, 29]. Though these methods have provided strong performance, a deficiency is that existing methods mostly transform the whole sequence into a fixed-length vector, which may introduce too much noise into the model learning process. Recently, some attention-based approaches are proposed [11, 27] to automatically assign different weights on items in terms of their relevance. However, these methods still cannot explicitly utilize the credible sequential patterns, thus the performance and interpretation are limited.

In light of these challenges mentioned above, we propose a **C**ontrastive **Lea**rning Model (named **CLEA**) to automatically extract the items relevant to the target item for next basket recommendation. The proposed CLEA consists of a sub-basket sequence generator, a context encoder, and a contrastive learning process for relevance modeling. The sequence generator works as a *denoising generator* (denoted as $\mathcal{G}$) with a Gumbel Softmax to explicitly extract the items relevant to the target item from each basket. In this sense, the denoising generator splits each basket into two disjoint sub-baskets: *positive* sub-basket and *negative* sub-basket. While a positive sub-basket contains all relevant items inside the corresponding basket, the negative sub-basket contains the other irrelevant ones instead. After this item-level denoising, we can form the two sub-basket sequences based on the output of the denoising generator. For each sub-basket sequence, we then derive the representation of each sub-basket based on its constituent items through a GRU-based context encoder, which expresses either relevant preference or irrelevant noises regarding the target item. Owing to the benefit of end-to-end training, our CLEA can automatically select items relevant to the target item and outperform the existing state-of-the-art recommendation models based on a two-stage anchor-guided contrastive learning process for model optimization.

To summarize, the contributions of this paper are listed as follows:

- We formulate a new task of performing denoising to enhance next basket recommendation. To the best of our knowledge, this is the first work of performing item-level denoising for a basket in an end-to-end fashion for this task.
- We propose a novel anchor-guided contrastive learning process to perform basket denoising without requiring any item-level relevance supervision, a two-stage learning approach to guarantee the effectiveness and efficiency of the learning procedure.
- Extensive experimental results over four real-world datasets show that our proposed CLEA can consistently outperform state-of-the-art baselines in terms of all Precision, Recall, F1, and NDCG metrics.

2 RELATED WORK

In this section, we briefly review two research areas related to our work, which are next basket recommendation and attention-based recommendation.

2.1 Next Basket Recommendation

Next basket recommendation is an important component for transaction data analysis. In previous decades, pattern mining-based methods are widely analyzed to mine insightful associations from transaction data [1]. This technique is known as association rule [2] and sequential pattern [31]. Many models are designed to mine qualified sequential patterns for recommendation. For example, Yap et al. [30] introduced a novel Competence Score measure to exploit user-specific sequential patterns for personalized recommendation. Wang et al. [28] proposed to address different types of noises as well as scalable algorithms developed for efficiently mining sequential patterns. Although simple and effective, these approaches can only provide recommendation based on the mined patterns, most of which are often noise once mining a large data set.

The MC-based methods are another line to model sequential properties.Previous work usually analyzes the impacts of single-step sequential behaviors. For example, Rendle et al. [19] designed a personalized Markov chain to provide recommendations. Based on this, Wang et al. [25] utilized representation learning metrics to model complex interactions between users and items.

Recently, the prosperous deep learning technology has begun to be applied in next basket recommendation task [3], which is used to fully model multi-step sequential behaviors. For example, Yu et al. [33] introduced an RNN-based approach to capture both user interests and global sequential features among baskets. Le et al. [14] incorporated information on pairwise correlations among items to enhance the representation of individual baskets. Wang et al. [27] built an attentive context to weight items with different relevance for recommendation. Besides, Leilei et al. [23] proposed a co-transformer multi-level representation method to capture the correlations of items and sets for sets prediction. Hu et al. [8, 9] designed a novel repeated purchase component on sequential models, and obtained considerable performance.

Though effective, these approaches fail to consider only the truly relevant items for the next choice, which complicates the preference learning and limits the performance and interpretability.

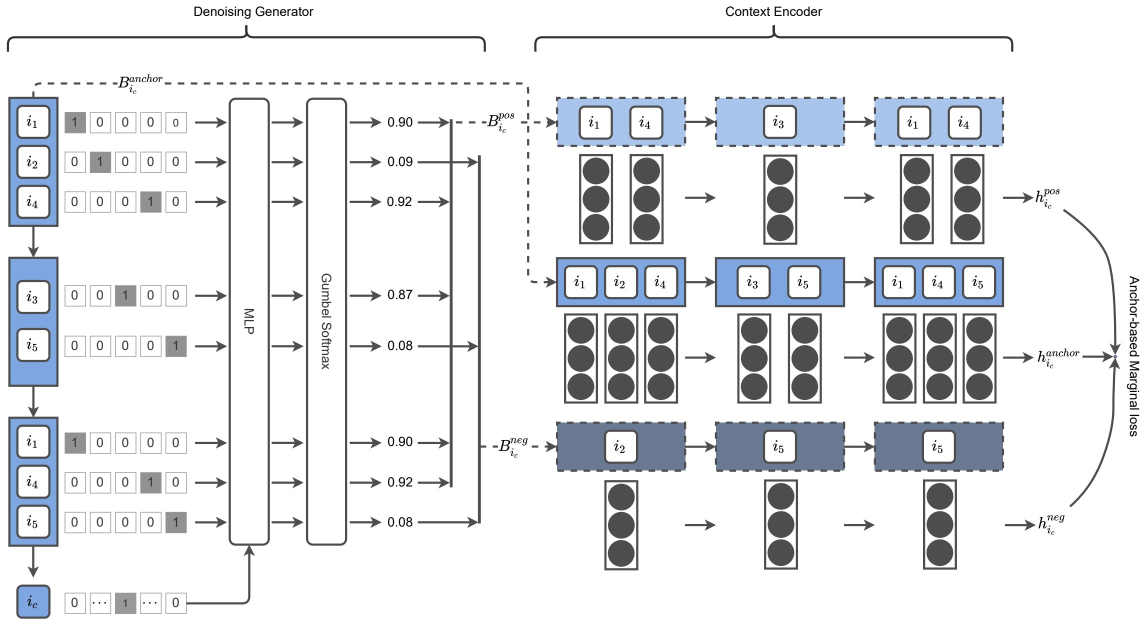

Figure 2: The network structure of the proposed CLEA.

2.2 Attention-based Recommendation

The attention mechanism allows a model to focus on the more informative parts of the target with different weights. Due to its promising performance in sequence learning, various models are designed for the sequential recommendation [15, 24].

For example, Ying et al. [32]proposed a two-layer hierarchical attention network to take both user-item and item-item interactions into account for sequential recommendation. SASRec [11] introduced a novel self-attention based sequential approach to model the entire user sequence, and adaptively considered consumed items for prediction. BERT4Rec [22] used a bidirectional self-attention network to model user sequential behaviors.

Different from our model, these attention-based models assign weights to softly weaken the adverse effects of the irrelevant items. However, this noisy information can still dilute the discriminative signals of the truly relevant ones. It is necessary to directly remove these irrelevant items for a clear recommendation. Recently, Zhang et al. [35] proposed a hierarchical reinforcement learning algorithm to automatically filter out the noisy courses. Different from their work, in this paper we address the denoising problem in a more complex basket scenario with a novel anchor-guided contrastive learning model.

3 OUR APPROACH

In this section, we first introduce the problem formulation of performing denoising for next basket recommendation. We then describe the proposed CLEA model in detail.

3.1 Problem Formulation

Without losing generality, we let $\mathcal{U} = \{u_1, u_2, \cdots, u_{|U|}\}$ denote all users and $\mathcal{I} = \{i_1, i_2, \cdots, i_{|I|}\}$ denote all items, where $|\mathcal{U}|$ and $|\mathcal{I}|$ represent the total number of unique users and items respectively. For each user $u \in \mathcal{U}$, we use $\mathcal{B}_u = \{b_u^1, b_u^2, \cdots, b_u^n\}$ to denote the recent purchase history of size n, where $b_u^t \in \mathcal{I}$ represents t-th basket containing the items purchased by the user at that time. Our aim is to extract a sub-basket sequence $\mathcal{B}_{u,i_c} = \{b_{u,i_c}^1, b_{u,i_c}^2, \cdots, b_{u,i_c}^n\}$ from $\mathcal{B}_u$, where b_{u,i_c}^t contains only the relevant items in b_u^t w.r.t. the target item i_c (*i.e.*, $b_{u,i_c}^t \subseteq b_u^t$), and precisely infer the likelihood that user u will purchase i_c at next visit based on $\mathcal{B}_{u,i_c}$.

For simplicity, we describe the technical details of CLEA for a single user u, and it is straightforward to extend the formulas to a set of users. Hence, we drop subscript u in the notations for concise presentation.

3.2 Denoising Generator

The whole architecture of CLEA is illustrated in Figure 2. Given a basket b_t and a target item i_c, we aim to design a denoising generator to automatically identify the items inside basket b_t that carry informative signals towards the user's preference over item i_c. Intuitively, the denoising generator can be considered as a binary classifier $y_{j,c} = \mathcal{G}(i_j, i_c)$ to decide whether item $i_j \in b_t$ is relevant to item i_c. When $y_{j,c}$=1, it means that i_j is relevant to i_c. In contrast, $y_{j,c} = 0$ indicates they are irrelevant. Formally, let $\mathbf{x}_j$ and $\mathbf{x}_c$ represent the one-hot representation of item i_j and i_c respectively, we can simply construct the denoising generator as a Multi-Layer Perception (MLP) to model the relevance between i_j and i_c:

$$\mathcal{G}(i_j, i_c) \sim f(\mathbf{x}_j \oplus \mathbf{x}_c) \tag{1}$$

where $\oplus$ is the vector concatenation operation, and MLP network $f(\cdot)$ with the sigmoid activation utilized in Eq. 1 performs nonlinear transfer to identify the relevance between the two items. Here, we can adopt a simple threshold or sampling strategy to binarize the real-valued output by $f(\cdot)$. For example, $y_{j,c}$ is set to 1 when $f(\mathbf{x}_j \oplus \mathbf{x}_c) \geq 0.5$.

However, this hard-coding (*i.e.,* 0 vs. 1) of item-level relevance by $\mathcal{G}$ is not differentiable and prevents the model from being trained well via back-propagation. To address this issue, inspired by [26, 34], we integrate a Gumbel Softmax into our denoising generator as a differentiable surrogate to support model learning over the discrete output. Specifically, the new denoising generator is rewritten as follows:

$$p = f(\mathbf{x}_j \oplus \mathbf{x}_c) \tag{2}$$

$$\mathcal{G}(i_j, i_c) = \frac{\exp((\log(p) + g_1)/\tau)}{\sum_{y=0}^{1} \exp(\log(p^y(1-p)^{1-y}) + g_y)/\tau)} \tag{3}$$

where g_y is i.i.d sampled from a Gumbel distribution as a noisy disturber: $g = -\log(-\log(x))$ and $x \sim \text{Uniform}(0, 1)$. In Eq. 3, $\tau > 0$ is the temperature parameter to smooth the discrete distribution of $\mathcal{G}$. That is, τ adjusts the sharpness of the relevance estimation generated by $\mathcal{G}$. When $\tau \rightarrow 0$, $\mathcal{G}$ approximates to a one-hot vector. On the other hand, when $\tau \rightarrow \infty$, $\mathcal{G}$ cannot distinguish the relevance from the counterpart.

Now, we can apply a simple threshold strategy to classify item i_j into postive sub-basket b_t^{pos} when $\mathcal{G}(i_j, i_c) \geq 0.5$; otherwise, negative sub-basket b_t^{neg}. In other words, we can explicitly decompose the purchase history of user u into a sequence of positive sub-baskets $B_{i_c}^{pos}=\{b_1^{pos}, b_2^{pos}, \cdots, b_n^{pos}\}$ and a sequence of negative sub-baskets $B_{i_c}^{neg}=\{b_1^{neg}, b_2^{neg}, \cdots, b_n^{neg}\}$.

3.3 Context Encoder

For each sub-basket, we firstly derive the embedding representation by utilizing a simple average pooling operation as follows:

$$\mathbf{b}_t^{pos} = \frac{1}{\sum_{i_j \in b_t^{pos}} \mathcal{G}(i_j, i_c)} \sum_{i_j \in b_t^{pos}} \mathcal{G}(i_j, i_c) \cdot \mathbf{v}_{i_j} \tag{4}$$

$$\mathbf{b}_t^{neg} = \frac{1}{\sum_{i_j \in b_t^{neg}} (1 - \mathcal{G}(i_j, i_c))} \sum_{i_j \in b_t^{neg}} (1 - \mathcal{G}(i_j, i_c)) \cdot \mathbf{v}_{i_j} \tag{5}$$

where $\mathbf{b}_t^{pos}$ and $\mathbf{b}_t^{neg}$ represent the positive and negative sub-basket representations of t-th basket respectively, and $\mathbf{v}_{i_j}$ is the learnable embedding representation for item i_j. Afterwards, it becomes straightforward to model the sequential interactions for two sub-basket sequences $\mathcal{B}_{i_c}^{pos}$ and $\mathcal{B}_{i_c}^{neg}$ respectively. Here, we utilize Gated Recurrent Unit (GRU) [4] as the context encoder to derive the hidden state for each sub-basket in the corresponding sequence[1]:

$$\mathbf{h}_t^{pos} = \text{GRU}(\mathbf{b}_t^{pos}, \mathbf{h}_{t-1}^{pos}) \tag{6}$$

The hidden state calculated for the last sub-basket b_n^{pos} (denoted as $\mathbf{h}_{i_c}^{pos}$) works as a composition of the user's preference towards target item i_c. On the contrary, the hidden state $\mathbf{h}_{i_c}^{neg}$ can be considered as a mixture of the user's preferences irrelevant to i_c. Note that the same GRU network is applied for both sub-basket sequences (*i.e.,* $\mathcal{B}_{i_c}^{pos}$ and $\mathcal{B}_{i_c}^{neg}$).

3.4 Anchor-Guided Contrastive Learning

The objective of model optimization is to guide the relevance learning of the denoising generator and help estimate the user's preference over the target item. As mentioned above, the preference feature $\mathbf{h}_{i_c}^{pos}$ is expected to precisely capture informative semantics that is relevant to the target item i_c. Hence, we would like to make a correct recommendation in terms of $\mathbf{h}_{i_c}^{pos}$. Actually, there are two correlated viewpoints to achieve this purpose: **(1)** each relevant item extracted by the denoising generator $\mathcal{G}$ indeed carries some discriminative signal towards the user's preference over item i_c against other items. In this sense, the target item should be easily inferred (*i.e.,* a high preference estimation) by performing the sequential modeling over the positive sub-baskets (*i.e.,* $\mathbf{h}_{i_c}^{pos}$); **(2)** each irrelevant item extracted by the denoising generator $\mathcal{G}$ should tell nothing towards the user's preference over item i_c. Hence, the preference estimation made in terms of $\mathbf{h}_{i_c}^{neg}$ should be relatively low.

Accordingly, we firstly calculate the likelihood that the user will purchase item i_c as follows:

$$\mathcal{D}(\mathbf{h}_{i_c}^{pos}) = \frac{exp(\mathbf{h}_{i_c}^{pos} \cdot \mathbf{w}_c)}{\sum_{i_l \in \mathcal{I}} exp(\mathbf{h}_{i_c}^{pos} \cdot \mathbf{w}_l)} \tag{7}$$

where $\mathbf{w}_c$ and $\mathbf{w}_l$ are the learnable preference embeddings for item i_c and i_l respectively, $\mathcal{D}(\mathbf{h}_{i_c}^{pos}) \in (0, 1)$ is the likelihood where a larger value indicates the higher preference. Then the objective function is formulated as follows:

$$\ell = \sum_{i_c \in b_u^{n+1}} \Big[log\big(\mathcal{D}(\mathbf{h}_{i_c}^{pos})\big) + log\big(1 - \mathcal{D}(\mathbf{h}_{i_c}^{neg})\big) \Big] - \lambda ||\Phi||^2 \tag{8}$$

where λ is the regularization coefficient. Symbol Φ represents all model parameters including Φ_G for denoising generator $\mathcal{G}$, Φ_{GRU} for GRU module and Φ_D for likelihood calculator $\mathcal{D}$. Many stochastic gradient descent techniques can be utilized to perform model learning. Here, we use the Adam optimizer [5] to maximize the objective function over all training instances. Also, to reduce the

Algorithm 1: Learning algorithm of CLEA

Input: learning rate α, iteration number *num*, number of epochs.
Output: Φ (Φ_{GRU}, $\Phi_{\mathcal{D}}$ and $\Phi_{\mathcal{G}}$)

Initialize $\Phi \leftarrow$ random values;
1: Stage 1 : the process of pretraining
for *epoch = 1 to EPOCHS1* **do**
  $\mathbf{b}_t \leftarrow \frac{1}{\sum_{i_j \in b_t}} \sum_{i_j \in b_t} \mathbf{v}_{i_j}$;
  $\mathbf{h}_t^{anchor} \leftarrow \text{GRU}(\mathbf{b}_t, \mathbf{h}_{t-1}^{anchor})$;
  $\mathbf{h}_{i_c}^{anchor} \leftarrow \mathbf{h}_T^{anchor}$;
  Train Φ_{GRU} and $\Phi_{\mathcal{D}}$ according Eq. 10;
2: Stage 2 :
while not converged **do**
  for *epoch = 1 to EPOCHS2* **do**
    Sample $g_y \sim$ Gumbel (0,1);
    Split B into $B_{i_c}^{pos}$ and $B_{i_c}^{neg}$ using Eq. 3;
    Update $\Phi_{\mathcal{G}}$ based on Eq. 9 with Φ_{GRU} and $\Phi_{\mathcal{D}}$ fixed;
    Anneal τ after a few batches;
  for *epoch = 1 to EPOCHS3* **do**
    Split B into $B_{i_c}^{pos}$ and $B_{i_c}^{neg}$ using Eq. 3;
    Update Φ_{GRU} and $\Phi_{\mathcal{D}}$ based on Eq. 9 with $\Phi_{\mathcal{G}}$ fixed;

[1] It is easy to utilize different sequence modeling networks here. We leave the exploration as a part of future work.

Table 1: Statistics of the datasets used in our experiments.

Dataset	#users	#items	#transactions	#average item count per transaction	#average transaction count per user
Ta-Feng	13,949	11,997	93,372	6.27	6.69
ValuedShopper	9,997	6,421	28,0762	9.17	28.08
Dunnhumby	36,241	4,995	176,849	10.80	5.00
Instacart	6,886	8,222	112,503	9.20	16.33

impact of model initialization and explore the parameters in a larger space, we start the model training with some relatively large τ for denoising generator $\mathcal{G}$, and then anneal it gradually to a small constant (ref. Section 4.1).

Note that the denoising generator $\mathcal{G}$ is expected to optimize the objective function by drawing a more precise distinction between the relevant items and the noise from a basket. However, this purpose is not guaranteed since no item-level relevance information is provided to supervise the learning of $\mathcal{G}$. In Eq. 8, we only calculate how accurate the proposed CLEA estimates the user's preference based on the training set. In the preliminary study, we find that the model training with Eq. 8 often produces unstable performance. As lack of supervised signals, the inferior learning of $\mathcal{G}$ could be easily repaired by the updating of the other parameters such as context encoder or preference embeddings (ref. Eq. 7).

Inspired by [21], we introduce a two-stage anchor-guided contrastive learning process to facilitate the optimization of CLEA. When denoising generator $\mathcal{G}$ performs under expectation, we can be certain about the user's preference over target item i_c in terms of $\mathbf{h}_{i_c}^{pos}$. However, when we turn off $\mathcal{G}$ by treating all items in each basket as relevant, the original sequence of historical baskets is equivalent to the corresponding positive sub-basket sequence. In this case, we can consider the resultant $\mathbf{h}_{i_c}^{pos}$ derived by the context encoder as a *preference anchor* $\mathbf{h}_{i_c}^{anchor}$ by setting $\mathcal{G}(i_j, i_c)$ to be 1 in Eq. 4. We believe that $\mathbf{h}_{i_c}^{anchor}$ could introduce some uncertainty, leading to a relatively low preference estimation, *i.e.,* $\mathcal{D}(\mathbf{h}_{i_c}^{pos}) > \mathcal{D}(\mathbf{h}_{i_c}^{anchor})$. Similarly, when $\mathcal{G}$ performs well, the negative sub-basket generated by $\mathcal{G}$ cannot infer the user's preference over item i_c. That is, the preference estimation made by $\mathbf{h}_{i_c}^{anchor}$ should be larger than the counterpart from $\mathbf{h}_{i_c}^{neg}$, *i.e.,* $\mathcal{D}(\mathbf{h}_{i_c}^{anchor}) > \mathcal{D}(\mathbf{h}_{i_c}^{neg})$. Therefore, this objective function can be written as follows:

$$\begin{aligned}\ell = \sum_{i_c \in b_u^{n+1}} \Big[& log\sigma\big(\mathcal{D}(\mathbf{h}_{i_c}^{pos}) - \mathcal{D}(\mathbf{h}_{i_c}^{anchor})\big) \\ & + log\sigma\big(\mathcal{D}(\mathbf{h}_{i_c}^{anchor}) - \mathcal{D}(\mathbf{h}_{i_c}^{neg})\big)\Big] - \lambda||\Phi||^2\end{aligned} \tag{9}$$

Where σ is a non-linear function which is chosen as $\sigma(x) = \frac{1}{1+e^{-x}}$. As we can see, comparing with Eq. 8, the newly added $\mathcal{D}(\mathbf{h}_{i_c}^{anchor})$ introduces a contrastive learning objective for $\mathcal{D}(\mathbf{h}_{i_c}^{pos})$ and $\mathcal{D}(\mathbf{h}_{i_c}^{neg})$ respectively. This forces CLEA to learn a robust $\mathbf{h}_{i_c}^{pos}$ to capture the credibly relevant items for each basket. Note that since both $\mathbf{h}_{i_c}^{pos}$ and $\mathbf{h}_{i_c}^{neg}$ are compared against anchor $\mathbf{h}_{i_c}^{anchor}$, the discriminative ability of $\mathbf{h}_{i_c}^{anchor}$ should be sufficiently strong to support the goal of denoising basket for better recommendation.

Therefore, we devise a two-stage learning process to accommodate the optimization of $\mathbf{h}_{i_c}^{anchor}$. Specifically, the whole margin learning process performs as follows:

- In the first step, we first train $\mathcal{D}(\mathbf{h}_{i_c}^{anchor})$ to obtain a benchmark without performing denoising generator, and the objective function is:

$$\ell_{anchor} = \sum_{i_c \in b_u^{n+1}} log\big(\mathcal{D}(\mathbf{h}_{i_c}^{anchor})\big) - \lambda||\Phi||^2 \tag{10}$$

- In the second stage, we use the well pretrained $\mathcal{D}(\mathbf{h}_{i_c}^{anchor})$ to guide the learning process of $\mathcal{D}(\mathbf{h}_{i_c}^{neg})$ and $\mathcal{D}(\mathbf{h}_{i_c}^{pos})$ according to Eq. 9. We iterate these two steps until coverage.

The detailed algorithm is shown in Algorithm 1. As for next basket recommendation, we use Eq. 3 to directly generate relevant items with threshold 0.5. Then, the learned likelihood calculator $\mathcal{D}$ is used to assign a likelihood score for each candidate item with Eq. 7. We then select the top-K items as the final recommendations.

4 EXPERIMENT

In this section, we conduct extensive experiments over four real-world datasets to evaluate the efficacy of CLEA against the up-to-date state-of-the-art alternatives.

4.1 Experimental Setup

Dataset. We use four real-world datasets in our experiments: TaFeng[2], ValuedShopper[3], Dunnhumby[4] and Instacart[5]. As for TaFeng, ValuedShopper and Dunnhumby, we take the corresponding datasets released in [9]. As for Instacart, it originally contains over 3 million grocery orders from more than 200, 000 users. We take all transactions of a sampled 10% users from the test type user set (*i.e.,* 75, 00 from 75000) for experiments. Following [13], we remove users and items with less than ten interaction records for all datasets. In addition, we filter out all the users with fewer than 3 transactions. The statistics of the four datasets after pre-processing are shown in Table 1.

For each user, we sort her/his records according to the timestamp to form the basket sequence. Based on the sorted baskets, we hold out the last basket of each user as test data, the penultimate one as validation data, and the rest as training data. Similar with [11], we utilize sampled metrics to speed up the computation. In order

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

Table 2: Performance comparison on basket recommendation between the baselines and our model with % omitted. The best and second best results in each row are highlighted in boldface and * respectively. ▲% refers to the absolute performance gain against the best baseline, which is consistently significant at 0.05 level.

Dataset	Evaluation Metric	Conventional Methods				Deep Methods						▲%
		POP-K	PSP	ItemKNN	FPMC	DREAM	GRU4Rec	SASRec	Beacon	BERT4Rec	CLEA	
Instacart	Recall@5	5.07	19.31	18.35	24.80	26.78	28.14	30.90	31.63	32.53*	**36.47**	3.94
	Prec@5	9.81	31.79	27.09	39.69	41.30	44.58	46.76	49.21	51.03*	**55.36**	4.33
	F1@5	6.11	21.35	19.19	27.19	28.68	30.54	33.07	34.20	35.35*	**39.08**	3.73
	NDCG@5	11.26	34.73	29.50	45.82	49.05	51.94	53.97	56.58	59.16*	**63.07**	3.91
Dunnhumby	Recall@5	5.40	8.39	8.08	10.21	11.14	11.36	12.08	12.68	13.71*	**15.34**	1.63
	Prec@5	22.00	34.45	27.86	41.38	42.41	43.75	46.36	49.57	51.38*	**59.03**	7.65
	F1@5	8.20	12.68	11.44	15.19	16.22	17.01	17.70	18.31	19.79*	**22.49**	2.70
	NDCG@5	21.75	39.21	23.11	44.07	47.41	48.62	50.57	54.29	56.54*	**63.08**	6.54
TaFeng	Recall@5	6.82	13.63	13.33	15.91	16.93	17.27	17.44	18.21	19.02*	**20.03**	1.01
	Prec@5	4.44	12.92	12.17	13.67	14.04	14.28	15.00	15.76	16.21*	**18.16**	1.95
	F1@5	4.31	11.06	10.52	11.88	12.46	12.68	12.91	13.74	14.29*	**15.56**	1.27
	NDCG@5	8.81	18.05	15.45	20.40	21.28	21.85	23.15	23.83	24.52*	**27.17**	2.65
Valuedshopper	Recall@5	5.70	9.88	8.67	12.30	12.95	14.33	14.70	15.43	16.48*	**17.73**	1.25
	Prec@5	21.25	36.67	30.75	46.76	47.74	51.37	53.02	55.99	58.48*	**63.43**	4.95
	F1@5	8.55	15.21	12.73	18.54	19.28	21.15	21.87	22.84	24.39*	**26.15**	1.76
	NDCG@5	23.42	38.77	29.95	49.95	51.57	55.75	57.07	60.73	62.86*	**66.10**	3.24

to improve the quality of the estimate, each ground-truth basket is paired with 1,000 negative items [12].

Baselines. Following [18], we compare the proposed CLEA against the conventional baselines and up-to-date deep models. ItemKNN and FPMC are implemented by RecBole [36].

- *Pop-K*: This is a weak baseline by returning top-K items from the training set in terms of basket frequency.
- *PSP* [31]: A personalized sequential pattern mining-based next item recommendation framework that exploits additional user-specific sequence importance.
- *ItemKNN*: [20]. It is a classical collaborative filtering solution based on item-level similarity, which is defined as the ratio of the co-occurrence number to the geometric mean of the two items in terms of basket frequency [18].
- *FPMC* [19]: FPMC is a shallow model that integrates hidden factors and first-order Markov chains together for basket recommendation.
- *DREAM* [33]: DREAM leverages recurrent neural network to model the dynamics of users' behaviors and the sequential patterns between items.
- *GRU4Rec* [7]: GRU4Rec models sequential information by Gated Recurrent Unit and the prediction for the next basket is made through a $softmax$ layer.
- *SASRec*[11]: SASRec utilizes a self-attention mechanism to model the whole sequence for sequential recommendation. To adapt SASRec for basket recommendation setting, the items in a basket share the same position embedding.
- *Beacon* [14]: It is a state-of-the-art next basket recommendation model that exploits pairwise item correlations.
- *BERT4Rec* [22]: BERT4Rec uses a bidirectional self-attention network to model user sequential behaviors. Similar to SASRec, we set the same position embedding for items of a basket to fit the next basket recommendation task.

Evaluation Metrics. For each user, a method will produce a top-K recommendation list for evaluation. We employ the widely used Precision@K (Prec@K), Recall@K, F1@K, and NDCG@K as evaluation metrics where K is set to 5 in our experiments. The statistical significance test is conducted by performing the paired *t-test*.

Parameter Settings. To make a fair comparison, we adopt the following settings for all methods: the batch size is set to 128; all embedding parameters are randomly initialized in the range of $(0, 1)$; and the model dimension is tuned in the range of $[32, 64, 96, 128, 256]$. We optimize each of them according to the validation sets. For ItemKNN, the number of nearest neighbors is set to 20. For PSP, we use PrefixSpan [17] to mine sequential patterns with the length up to 6, the mining support is tuned in the range of [0.01,0.05,0.1,0.2,0.5]. For SASRec, we set the maximum length of a basket sequence to 20 (*i.e.,* $n = 20$), and we add paddings when the sequence length is fewer than that. For our model, the hidden dimension and embedding size are both set to 64. We optimize our model with adam optimizer. The initial temperature τ in Gumbel Softmax is set to 10. We follow [10] to anneal τ according to the following schedule:

$$\tau \leftarrow max\big(0.3, \tau exp(-rt)\big)$$

where t is the global training batches, r represents the decay rate, which is set to $r = 10^{-4}$. We anneal the temperature according the above function after every 500 batches.

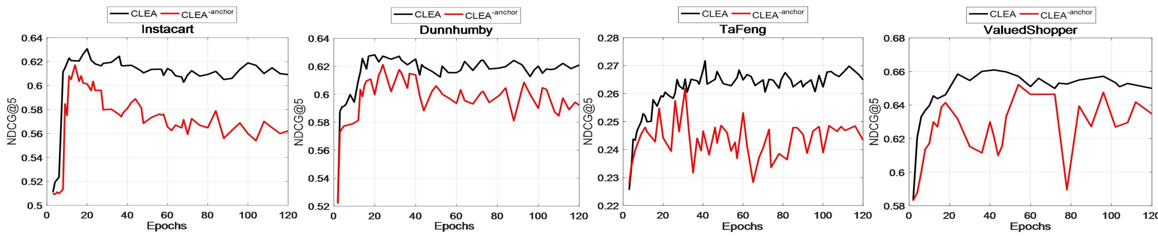

Figure 3: Performance curves of CLEA and CLEA $^{-anchor}$ with varying iteration numbers.

4.2 Performance Comparison

Table 2 presents a summary of experimental results of different methods. Here, we have the following observations:

(1) For conventional methods, it is expected that Pop-K performs worst on all datasets. Both PSP and ItemKNN perform better than Pop-K by considering interactions among items. Comparing with ItemKNN, PSP can capture high-order sequential patterns, resulting in better performance on all datasets. Though effective, PSP and ItemKNN face the sparse problem as these statistical approaches require sufficient data. By modeling both user's general taste and sequential patterns, FPMC alleviates the sparse problem and performs better than other conventional methods.

(2) It is clear that deep methods perform better than conventional methods on all datasets. This verifies the necessity of applying nonlinear learning to model the complex interactions among items. Among these models, Beacon utilizes correlation information over items to enhance the representations of basket sequences, and outperforms DREAM and GRU4Rec, which equally treat all items' contributions in sequential models. In addition, we observe that SASRec outperforms DREAM and GRU4Rec on the four datasets too, which indicating that self-attention mechanism is effective for sequential modeling. It is worthwhile to highlight that BERT4Rec > Beacon > SASRec. The reason might be that bidirectional self-attention network and the Cloze objective are more effective to model sequential information, which is confirmed in [22].

(3) Finally, our proposed CLEA achieves the best performance against all the methods on the four datasets. Though Beacon also suggests that there exists informative correlations that are beneficial for next basket recommendation, it cannot identify which item is really needed for estimating the next choice. Besides, the irrelevant noises are also kept, which may complicate the preference learning. The core novelty of CLEA is to automatically analyze interactions between historical items and the target item. Comparing with Beacon, CLEA can well model these interactions from a more microscopic perspective. Hence, a more precise estimation of the user's preference can be learnt. Specifically, taking the Instacart dataset as an example, when compared with the best baseline method (*i.e.,* BERT4Rec), the absolute performance improvement by CLEA is around 3.94%, 4.33%, 3.73% and 3.91% in terms of Precision@5, Recall@5, F1-score, and NDCG@5 respectively.

4.3 Further Analysis

Analysis on Different User Groups. To further investigate the performance of different methods, we split the users into three groups (i.e., sparse, medium and dense) based on their average basket size, and conducted the comparisons on different user groups. Take Instacart dataset as an example, a user is classified into the sparse group if the average basket size is less than 5, and dense if it is larger than 10. The remaining users are taken as the medium. In this way, the proportions of sparse, medium and dense are 20%, 45%, and 35% respectively. Here we only report the comparison results on Instacart dataset, similar conclusions can also be drawn from other datasets. The results are shown in Table 3. From the results, we can see that CLEA consistently achieves significant performance gain against all the baselines, including the attention-based solutions like SASRec and BERT4Rec. This further validates the effectiveness of CLEA to perform basket denoising for better recommendation performance.

Table 3: Performance comparison on Instacart over different user groups. The * indicates the best performed in (a)-(i). Best performance is written in bold.

User Groups		Method	Recall@5	Prec@5	F1@5	NDCG@5
Sparse	(a)	POP-K	4.32	4.55	4.12	5.45
	(b)	PSP	29.31	22.95	22.81	32.17
	(c)	ItemKNN	27.57	19.92	20.89	27.07
	(d)	FPMC	33.84	27.42	27.86	39.05
	(e)	DREAM	35.73	29.32	31.07	41.66
	(f)	GRU4Rec	39.99	31.37	31.92	46.70
	(g)	Beacon	41.31	31.66	32.39	47.44
	(h)	SASRec	39.38	30.72	31.33	45.82
	(i)	BERT4Rec	41.67*	32.39*	33.06*	48.66*
	(j)	CLEA	**45.61**	**36.40**	**36.82**	**53.05**
	▲%		3.94	4.01	3.76	4.39
Medium	(a)	POP-K	5.09	8.16	5.82	9.59
	(b)	PSP	20.01	30.11	22.27	33.80
	(c)	ItemKNN	18.54	25.50	19.77	27.55
	(d)	FPMC	25.17	36.75	27.81	41.85
	(e)	DREAM	26.88	38.94	27.99	44.70
	(f)	GRU4Rec	29.14	42.36	31.81	49.03
	(g)	Beacon	32.78	46.20	35.67	54.05
	(h)	SASRec	31.15	43.30	33.96	51.17
	(i)	BERT4Rec	33.00*	46.84*	35.89*	54.41*
	(j)	CLEA	**37.68**	**52.42**	**40.63**	**60.08**
	▲%		4.68	5.58	4.74	5.67
Dense	(a)	POP-K	5.49	14.98	7.65	15.76
	(b)	PSP	13.14	38.62	19.02	38.43
	(c)	ItemKNN	12.81	33.26	17.46	33.42
	(d)	FPMC	18.65	49.53	24.99	52.18
	(e)	DREAM	19.92	49.87	26.51	54.13
	(f)	GRU4Rec	20.43	54.18	28.16	57.94
	(g)	Beacon	24.60	63.01	33.49	65.82
	(h)	SASRec	23.91	60.48	32.12	63.69
	(i)	BERT4Rec	26.61*	67.12*	35.96*	71.05*
	(j)	CLEA	**28.63**	**70.09**	**38.51**	**73.11**
	▲%		2.02	2.97	2.55	2.06

Ablation Study. The core merit of CLEA is that a denoising generator and a contrastive learning process $\mathcal{G}$ are devised to utilize only items relevant to the target item for preference learning. Here,

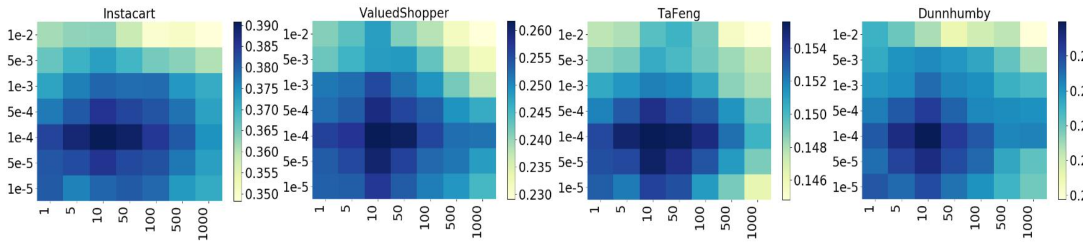

Figure 4: Heatmap on four datasets. The x-axis denotes the initial temperature in $\{1, 5, 10, 100, 500, 1000\}$, and y-axis denotes different decay rates. A darker color represents a better performance.

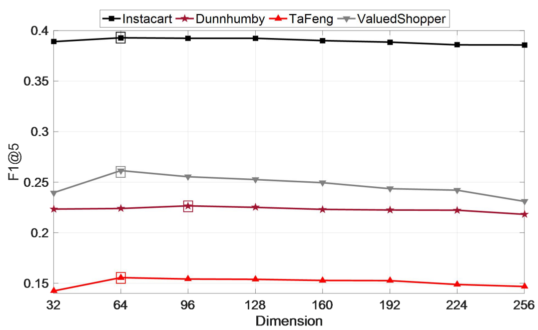

Figure 5: The impact of the dimension size in terms of F1 on four datasets. The dimensionality is increased from 32 to 256. The rectangle indicates the best performance on the dataset.

Table 4: Performance comparison of three different variation models against CLEA.

Dataset	Evaluation Metric	CLEA	CLEA^{-den}	CLEA^{soft}	CLEA$_{jot}$
Instacart	Recall@5	**36.47**	31.26	34.53	35.24
	Precision@5	**55.36**	48.69	53.11	54.34
	F1@5	**39.08**	33.77	37.04	38.16
	NDCG@5	**63.07**	56.10	60.76	62.10
Dunnhumby	Recall@5	**15.34**	12.03	13.80	14.88
	Precision@5	**59.03**	46.97	54.27	57.77
	F1@5	**22.49**	17.70	20.36	21.77
	NDCG@5	**63.08**	52.19	59.08	61.63
TaFeng	Recall@5	**20.03**	17.93	18.55	18.77
	Precision@5	**18.16**	15.80	16.23	16.35
	F1@5	**15.56**	13.61	14.04	14.21
	NDCG@5	**27.17**	23.88	24.68	25.33
ValuedShopper	Recall@5	**17.73**	14.88	16.12	16.01
	Precision@5	**63.43**	53.92	59.34	58.28
	F1@5	**26.15**	22.16	24.27	23.76
	NDCG@5	**66.10**	58.31	63.01	62.07

we perform a series of ablation studies to check the effect of each design choice with the performance difference over the test set.

In Eq. 9, we utilize the original basket sequence as an anchor to guide both the relevance and preference learning. Hence, we first investigate the effect of this contrastive learning process by removing this anchor-guided objective. Instead, we use Eq. 8 for model optimization. We name this variant as CLEA $^{-anchor}$. Figure 3 plots the performance curves of CLEA and CLEA $^{-anchor}$. We can see that comparing with CLEA $^{-anchor}$, CLEA obtains a better performance and converges faster. These observations on all four datasets are quite consistent. It demonstrates the correctness of the anchor-guided contrastive learning devised in CLEA.

Moreover, we remove the denoising generator $\mathcal{G}$ on the basis of CLEA. In this sense, the original basket sequence $\mathcal{B}_u$ was no longer divided into two sub-baskets, but is directly fed into the context encoder to obtain its aggregated representation h^{anchor}. We name this degraded model as CLEA $^{-den}$, and perform optimization according to Eq. 10. Comparing with the previous attention methods that assign weights on items, CLEA utilizes a hard-coding scheme to check whether each item is relevant to the target item or not. Specifically, when obtaining embeddings of sub-baskets in Eq. 4 and 5, we choose not to split the original basket into two disjoint sub-baskets. That is, we only depend on the output of $\mathcal{G}(i_j, i_c)$ as an attention mechanism to derive the sub-basket representations. We name this variant as CLEA ^{soft}. Moreover, we also examine the effect of the two-stage learning by performing a joint learning instead (namely CLEA ^{jot}). The results of CLEA and its three variants models are shown in Table 4.

We can see that CLEA $^{-den}$ performs worst on all evaluation metrics over four datasets. It indicates the necessity of dropping irrelevant information to enhance preference learning. When applying a soft attention on items, CLEA ^{soft} can weaken the importance of irrelevant items, and performs better than CLEA $^{-den}$. This also coincides with the previous findings in [11]. Also, in contrast to CLEA ^{soft}, CLEA uses a hard-coding to directly drop these irrelevant noises, and shows a better performance than CLEA ^{soft}. At last, we can observe that the two-stage learning process utilized in CLEA can produce better recommendation performance (*i.e.*, CLEA vs. CLEA ^{jot}). This also verifies the importance of learning a robust anchor to guide the basket denoising. In summary, this set of experimental comparisons suggests that each design choice in CLEA is rational to enhance next basket recommendation via basket denoising.

Analysis on Gumble Softmax. Recall that we use Gumble Softmax to facilitate the modeling learning for CLEA. The temperature τ and decay rate γ are two important hyperparameters. Here, we examine different choices of τ and γ on four datasets, and analyze their impacts to the recommendation performance. The results on four datasets are shown in Figure 4.

As we can see, given a fixed decay rate γ, when τ increases, the performance of CLEA increases too. This observation demonstrates

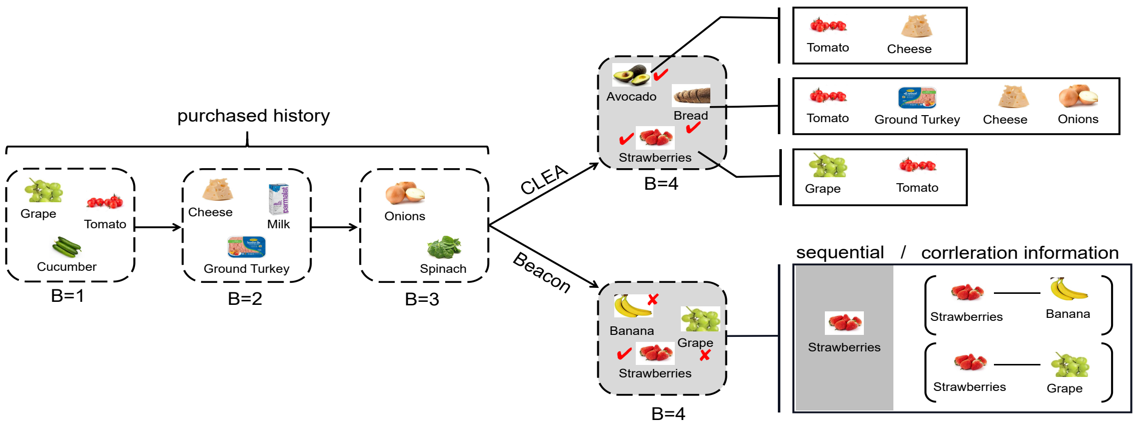

Figure 6: Case study on next-basket recommendation task. The left row is a basket sequence of a user sampled from Instacart dataset. The right two rows are recommendation results of CLEA and Beacon respectively. Items and patterns in boxes indicate relevant information identified by CLEA and Beacon respectively.

that we need a large initial temperature to guarantee a broader sampling range for Eq. 3. On the other hand, when we continue to increase the initial temperature, the overall performance decreases. The reason is that a higher temperature exacerbates the difficulty of model convergence. Similar observations are also made for γ. Specifically, a small γ (i.e., $\gamma = 10^{-5}$) makes CLEA difficult to converge, and a high γ (i.e., $\gamma = 10^{-2}$) limits the ability of exploring in a broader sampling space. Based on the results, we set the initial temperature as 10 and decay rare γ as 10^{-4} in our experiments.

Analysis on Embedding Size. At last, we study the impact of different embedding sizes in the range of 32 to 256. The performance patterns on the four datasets are plotted in Figure 5. We see that CLEA achieves optimal performance with a small dimension on all four datasets (*i.e.,* 64, 64, 96 and 64 on Instacart, ValuedShopper, Dunnhumby and TaFeng respectively). Also, the performance decreases consistently as the embedding size becomes increasingly large. We believe that since the denoising process removes many irrelevant items, there is no need to model the complex interactions between items with a large embedding. Note that a model with a large embedding size would also easily fall into overfitting.

4.4 QUALITATIVE ANALYSIS

Previous experiments have well demonstrated that the proposed CLEA is effective to improve the next basket recommendation performance. In order to better understand why it is useful, we further perform qualitative analysis with a case study on Instacart dataset. Specifically, as shown in Figure 6, we present a snapshot of the basket sequence for a sample user. To ease the illustration we only display the last 3 baskets. Given the basket sequence, we use CLEA and Beacon to recommend what the user will buy next. For a convenient demonstration, we provide top-3 recommendation results of two models.

We can see that CLEA can recommend all 3 items correctly. For example, when determining whether *strawberries* will be preferred, CLEA identifies $\{grape, tomato\}$ as the relevant items, and produces a high likelihood score according to Eq. 7. Similar decision making process is also observed on *bread* and *avocado*, where CLEA finds $\{tomato, turkey, cheese, onion\}$ and $\{tomato, cheese\}$ as two important patterns respectively. In contrast, by applying the global correlation analysis with *strawberries*, Beacon considers *banana* and *grape* as another two optimal recommendations. As we can see, by generating the positive sub-basket sequence through the denoising generator, our CLEA can well capture diverse user preferences, and in turn make a more precise recommendation.

We also count the irrelevant items that CLEA removed. Specifically, the average removed items for each basket are 5.01, 7.54, 8.92, and 6.23 on TaFeng, Instcacart, Dunnhumby, and ValuedShopper respectively. It demonstrates that CLEA does denoise items from baskets. By dropping these irrelevant items, CLEA obtains superior performance on all datasets.

5 CONCLUSION

In this paper, we address a denoising problem in next basket recommendation scenario, we propose a **C**ontrastive **Lea**rning Model (named **CLEA**) to automatically extract items relevant to the target item for next basket recommendation. Specifically, we first devise a denoising generator to adaptively identify whether each item in a historical basket is relevant to the target item or not. By splitting the initial basket into two sub-baskets, we derive their representations through a GRU-based context encoder, which expresses either relevant interest or irrelevant noises regarding the target item. After that, a novel anchor-guided contrastive learning process is then designed to simultaneously guide this relevance learning without requiring any item-level relevance supervision. To the best of our knowledge, this is the first work of performing item-level denoising for a basket in an end-to-end fashion for next basket recommendation. As future work, we plan to exploit weakly supervised signals to better understand interactions among items for further improvement.

ACKNOWLEDGMENTS

This research work was supported by fundamental Research for the National Natural Science Foundation of China (No.61802029,61872278). We would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N Swami. 1993. Mining association rules between sets of items in large databases. 22, 2 (1993), 207–216.

[2] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile.* 487–499.

[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, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.* 1201–1204.

[4] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL.* 1724–1734.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada.* 2672–2680.

[6] R. Guidotti, G. Rossetti, L. Pappalardo, F. Giannotti, and D. Pedreschi. 2019. Personalized Market Basket Prediction with Temporal Annotated Recurring Sequences. *IEEE Transactions on Knowledge and Data Engineering* 31, 11 (2019), 2151–2163.

[7] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.*

[8] 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 & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 1491–1499.

[9] 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020.* 1071–1080.

[10] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.*

[11] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018.* 197–206.

[12] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020.* 1748–1757.

[13] Duc-Trong Le, Hady Wirawan Lauw, and Yuan Fang. 2017. Basket-Sensitive Personalized Item Recommendation. In *IJCAI 19-25, 2017.* 2060–2066.

[14] Duc-Trong Le, Hady W. Lauw, and Yuan Fang. 2019. Correlation-Sensitive Next-Basket Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019.* 2808–2814.

[15] Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020.* 322–330. https://doi.org/10.1145/3336191.3371786

[16] Anjing Luo, Pengpeng Zhao, Yanchi Liu, Fuzhen Zhuang, Deqing Wang, Jiajie Xu, Junhua Fang, and Victor S. Sheng. 2020. Collaborative Self-Attention Network for Session-based Recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 2591–2597. https://doi.org/10.24963/ijcai.2020/359

[17] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei chun Hsu. 2001. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. 215–224.

[18] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.* 4806–4813.

[19] 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 - WWW '10.* ACM Press, New York, New York, USA, 811–820. https://doi.org/10.1145/1772690.1772773

[20] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001.* 285–295.

[21] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015.* 815–823.

[22] 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, CIKM 2019, Beijing, China, November 3-7, 2019.* 1441–1450.

[23] 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020.* 1439–1448.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA.* 5998–6008.

[25] 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, Santiago, Chile, August 9-13, 2015.* 403–412.

[26] Qinyong Wang, Hongzhi Yin, Hao Wang, Quoc Viet Hung Nguyen, Zi Huang, and Lizhen Cui. 2019. Enhancing Collaborative Filtering with Generative Augmentation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 548–556.

[27] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-Based Transactional Context Embedding for Next-Item Recommendation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018.* 2532–2539.

[28] Wei Wang, Jiong Yang, and Philip S. Yu. 2000. Mining Patterns in Long Sequential Data with Noise. *SIGKDD Explorations* 2, 2 (2000), 28–33.

[29] Chengfeng Xu, Jian Feng, Pengpeng Zhao, Fuzhen Zhuang, Deqing Wang, Yanchi Liu, and Victor S. Sheng. 2021. Long- and short-term self-attention network for sequential recommendation. *Neurocomputing* 423 (2021), 580 – 589.

[30] Ghim-Eng Yap, Xiao-Li Li, and Philip S. Yu. 2012. Effective Next-Items Recommendation via Personalized Sequential Pattern Mining. In *International Conference on Database Systems for Advanced Applications.*

[31] Ghim Eng Yap, Xiao Li Li, and Philip S. Yu. 2015. Effective Next-Items Recommendation via Personalized Sequential Pattern Mining. In *International Conference on Database Systems for Advanced Applications.*

[32] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.* 3926–3932.

[33] 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, SIGIR 2016, Pisa, Italy, July 17-21, 2016.* 729–732.

[34] Junliang Yu, Min Gao, Hongzhi Yin, Jundong Li, Chongming Gao, and Qinyong Wang. 2019. Generating Reliable Friends via Adversarial Training to Improve Social Recommendation. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019.* 768–777.

[35] Jing Zhang, Bowen Hao, Bo Chen, Cuiping Li, Hong Chen, and Jimeng Sun. 2019. Hierarchical Reinforcement Learning for Course Recommendation in MOOCs. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.* 435–442.

[36] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2020. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. *arXiv preprint arXiv:2011.01731* (2020).