A Next Basket Recommendation Reality Check

MING LI, SAMI JULLIEN, and MOZHDEH ARIANNEZHAD, AIRLab, University of Amsterdam
MAARTEN DE RIJKE, University of Amsterdam

The goal of a next basket recommendation (NBR) system is to recommend items for the next basket for a user, based on the sequence of their prior baskets. We examine whether the performance gains of the NBR methods reported in the literature hold up under a fair and comprehensive comparison. To clarify the mixed picture that emerges from our comparison, we provide a novel angle on the evaluation of next basket recommendation (NBR) methods, centered on the distinction between repetition and exploration: the next basket is typically composed of previously consumed items (i.e., repeat items) and new items (i.e., explore items). We propose a set of metrics that measure the repetition/exploration ratio and performance of NBR models. Using these new metrics, we provide a second analysis of state-of-the-art NBR models. The results help to clarify the extent of the actual progress achieved by existing NBR methods as well as the underlying reasons for any improvements that we observe. Overall, our work sheds light on the evaluation problem of NBR, provides a new evaluation protocol, and yields useful insights for the design of models for this task.

CCS Concepts: • **General and reference** → Evaluation; • **Information systems** → **Recommender systems**;

Additional Key Words and Phrases: Next basket recommendation, reproducibility, repeat behavior

1 INTRODUCTION

Over the years, **Next Basket Recommendation (NBR)** has received a considerable amount of interest from the research community [6, 33, 43]. Baskets, or sets of items that are purchased or consumed together, are pervasive in many real-world services, with e-commerce and grocery shopping as prominent examples [18, 32]. Given a sequence of baskets that a user has purchased or consumed in the past, the goal of an next basket recommendation (NBR) system is to generate the basket of items that the user would like to purchase or consume next. Within a basket, items

have no temporal order and are equally important. A key difference between NBR and session-based or sequential item recommendations is that NBR systems need to deal with multiple items in one set. Therefore, models designed for item-based recommendation are not fit for basket-based recommendation, and dedicated NBR methods have been proposed [16, 20, 21, 25, 31, 34, 41, 43].

1.1 Types of Recommendation Methods

Over the years, we have seen the development of a wide range of recommendation methods. *Frequency-based* methods continue to play an important role, as they are able to capture global statistics concerning popularity of items; this holds true for item-based recommendation scenarios as well as for NBR scenarios. Similarly, *nearest neighbor based* methods have long been used for both item-based and basket-based recommendation scenarios. More recently, deep learning techniques have been developed to address sequential item recommendation problems, building on the capacity of deep learning based methods to capture hidden relations and automatically learn representations [7]. Recent years have also witnessed proposals to address different aspects of the NBR task with deep learning based methods, such as item-to-item relations [25], cross-basket relations [44], and noise within a basket [31].

Recent analyses indicate that deep learning based approaches may not be the best performing for all recommendation tasks and under all conditions [22]. For the task of generating a personalized ranked list of items, linear models and nearest neighbor based approaches outperform deep learning based methods [14]. For sequential recommendation problems, deep learning based methods may be outperformed by simple nearest neighbor or graph-based baselines [15]. What about the task of NBR? Here, the unit of retrieval—a basket—is more complex than in the recommendation scenarios considered in other works [14, 15, 22], with complex dependencies between items and baskets, across time, thus creating a potential for sophisticated representation learning based approaches to NBR to yield performance gains. In this article, we take a closer look at the field to see if this is actually true.

1.2 A New Analysis Perspective

We find important gaps and flaws in the literature on NBR. These include weak or missing baselines, the use of different datasets in different papers, and of non-standard metrics. We evaluate the performance of three families of state-of-the-art NBR models (frequency based, nearest neighbor based, and deep learning based) on three benchmark datasets, and find that no NBR method consistently outperforms all other methods across all settings.

Given these outcomes, we propose a more thorough analysis of the successes and failures of NBR methods. As we show in Figure 1, baskets recommended in an NBR scenario consist of *repeat items* (items that the user has consumed before, in previous baskets) and *explore items* (items that are new to the user). The novelty of recommended items has been studied before, and related metrics have also been proposed [36], but novelty-oriented metrics are not NBR specific and only focus on one aspect (i.e., evaluating the novelty of the list of recommendations). To improve our understanding of the relative performance of NBR models, especially regarding repeat items and explore items, we introduce a set of task-specific metrics for NBR. Our newly proposed metrics help us understand which types of items are present in the recommended basket and assess the performance of NBR models when proposing new items vs. already-purchased items.

1.3 Main Findings

Equipped with our newly proposed metrics for NBR, we repeat our large-scale comparison of NBR models and arrive at the following important findings:
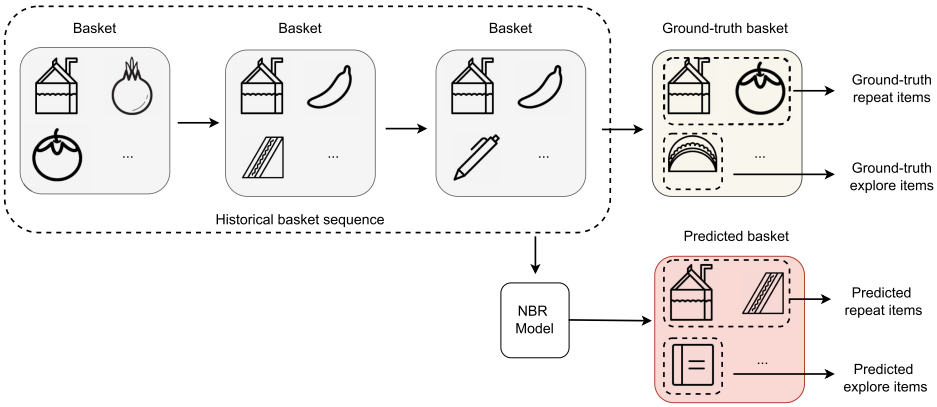
Fig. 1. Four types of items in NBR.

- No NBR method consistently outperforms all other methods across different datasets.
- All published methods are heavily skewed toward either repetition or exploration compared to the ground truth, which might harm long-term engagement.
- There is a large NBR performance gap between repetition and exploration; repeat item recommendation is much easier.
- In many settings, deep learning based NBR methods are outperformed by frequency-based baselines that fill a basket with the most frequent items in a user's history, possibly complemented with items that are most frequent across all users.
- A bias toward repeat items accounts for most of the performance gains of recently published methods, even though many complex modules or strategies specifically target explore items.
- We propose a new protocol for evaluating NBR methods, with a new frequency-based NBR baseline as well as new metrics to assess the potential performance gains of NBR methods.
- Existing NBR methods have different treatment effects on user performance and item exposure for users with different repetition ratios and items with different frequencies, respectively.

Overall, our work sheds light on the state of the art of NBR, provides suggestions to improve our evaluation methodology for NBR, helps us understand the reasons underlying performance differences, and provides insights to inform the design of future NBR models.

2 RELATED WORK

2.1 Reproducibility in Information Retrieval

Reproducibility is a topic that has been at the center of Information Retrieval (IR) research for many years. The *mechanics* of reproducibility have been a constant factor since the early days of community benchmarking [37], resulting in a large number of datasets and metrics. Artifact badging is a matter of ongoing and active interest [17], as are ways to objectively quantify to what extent a system-oriented information retrieval (IR) experiment has been replicated or reproduced [9].

Asking *which lessons hold up* under closer scrutiny is not new either in IR. Papers of this type have been written for query performance prediction [19], ranking [5], learning to rank [35], search result diversification [1], online learning to rank [29], question answering [13], and neural rankers [28, 42]. We are particularly interested in this "which lessons hold up" aspect of reproducibility in the context of recommender systems. Dacrema et al. [14, 15] and Jannach et al. [22] have recently examined the relative strength of deep learning based methods for item

recommendation, both in a traditional static setting and in a sequential setting. In contrast, we consider the task of NBR and assess the relative merits of deep learning based methods for this task.

Some NBR methods [6, 25, 31, 41, 43] only compare with previous (deep) learning based methods and avoid comparing with frequency-based baselines that recommend the k most frequent items in a users' historical records as the next basket. In several cases, recent publications on NBR omit comparisons to other recent methods [16, 21, 31, 44]. In the work of Qin et al. [31], sampled metrics [23] are used to evaluate the performance even though this is not encouraged [24]. As the following systematic comparisons show, not all previously published lessons on deep learning based NBR methods hold up.

2.2 Next Basket Recommendation

The NBR problem has been studied for many years. As we explain in Section 3 below, we analyze the performance of three families of NBR methods. First, we consider frequency-based methods; in different configurations they have been considered as baselines in most prior work on NBR that we are aware of. Second are nearest neighbor-based methods. TIFUKNN [21] and UP-CF@r [16] model temporal patterns over frequency information and then combine with neighbor information or user-wise collaborative filtering. Third are deep learning-based methods. Such methods often have a strong focus on learning representations of baskets. Early precursors include the Factorizing Personalized Markov Chains (FPMC) [33], which leverage matrix factorization and Markov chains to model users' general interest and basket transition relations, and hierarchical representation models [39], which apply aggregation operations to learn a hierarchical representation of baskets; these two Markov chain based methods only capture local short-term relations between adjacent baskets. In contrast, RNNs have been used for the NBR task to learn long-term trends by modeling the whole basket sequence. For instance, DREAM [43] uses max/avg pooling to encode baskets. Sets2Sets [20] adapts an attention mechanism and adds frequency information to improve performance. Some methods [25, 41] consider item relations to obtain a better representation. Yu et al. [44] argue that item-item relations between baskets are important and leverage GNNs to capture these relations; the authors also use a self-attention mechanism to learn temporal dependencies between baskets. Some methods [6, 12, 26, 34, 40] exploit auxiliary information, including product categories, amounts, prices, and explicit timestamps; for the sake of a fair comparison, we omit these from our reproducibility study.

What we add on top of prior work is not yet another NBR method but a systematic comparison under the same experimental conditions across multiple datasets as well as an analysis of the relative performance of state-of-the-art NBR methods in terms of repetition and exploration, which helps to explain the observed performance differences.

3 DATASETS AND METHODS

3.1 Datasets

To ensure the reproducibility of our study, we conduct our experiments on three publicly available real-world datasets:

- *TaFeng* contains 4 months of shopping transactions collected from a Chinese grocery store. All products purchased on the same day by the same user are treated as a basket.[1]
- *Dunnhumby* covers 2 years of household-level transactions at a retailer. All products bought by the same user in the same transaction are treated as a basket. We use the first 2 months of the data in our experiments.[2]

---

[1]https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset.
[2]https://www.dunnhumby.com/source-files/.

Table 1. 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

- *Instacart* contains more than 3 million grocery orders of Instacart users. We treat all items purchased by the same user in the same order as a basket.[3]

Following previous work [20, 21, 25, 43, 44], we also employ a sampling strategy instead of using the whole dataset. In each dataset, users with a basket size between 3 and 50 are sampled to conduct experiments. We also remove rare and unpopular items, and the remainder covers more than 95% of the interactions. A *ground-truth basket* is a basket that we aim to predict or recommend, and the last basket of a sequence or purchased baskets is regarded as the ground-truth basket. The *repetition ratio* and *exploration ratio* are calculated based on the ground-truth basket as the proportion of *repeat items* and *explore items* in the ground-truth baskets, respectively. The statistics of the processed datasets are summarized in Table 1.

For our experiments, we split every dataset across users like previous works [20, 21, 44] (i.e., 72% users for training, 8% users for validation, and 20% users for the test set). The training users, validation users, and test users are totally independent from each other. We repeat the dataset split five times for independent experiments. Note that we did not use an absolute timestamp splitting strategy (i.e., splitting the dataset into several sub-datasets according to real time ranges) for the following reasons:

(1) As a reality check paper, we decided to follow the widely used data splitting strategy in the existing NBR methods.
(2) We checked three datasets' characteristics and found that users' baskets are spread out across diverse time ranges and the average basket lengths are limited, so splitting using an absolute timestamp would be likely to break a large number of basket sequences into very short sequences, which cannot be used for effectively training or evaluating NBR methods.
(3) As the article focuses on the analysis from a repetition and exploration perspective, we acknowledge that global trends are likely to influence the repetition ratio of the users, and therefore we use Section 5.5 to analyze at a fine-grained level (i.e., the treatment on users with different repetition ratios).

3.2 Baseline Methods and Reproducible Methods

We follow the same strategy as Dacrema et al. [15] to collect relevant and reproducible NBR papers. Specifically, we include papers in our analysis that were published from 2016 to 2021 in one of the following conferences: KDD [20, 34, 44], SIGIR [6, 21, 43], IJCAI [25], AAAI [41], RecSys [12], and UMAP [16]. Papers targeting NBR [6, 16, 21, 25, 31, 41, 43] and sequential set prediction [20, 34, 44] are considered to be relevant papers. For a fair comparison, methods [6, 12, 34] using auxiliary information other than item-basket sequences are not included in this work. Like Dacrema et al. [15], we consider a paper to be reproducible if it meets the following criteria:

---

[3]https://www.kaggle.com/c/instacart-market-basket-analysis/data.

Table 2. Percentage of the Basket Slots Used for
P-TopFreq

Basket Size	Dataset		
	TaFeng	Dunnhumby	Instacart
10	92.39%	95.66%	96.70%
20	79.71%	87.98%	90.48%

(1) A working version of the *source code* is publicly available[4] or the code has to be modified in minimal ways to work correctly.[5]
(2) At least one *dataset* used in the original paper is available.

Through this selection process, we end up with eight relevant representative papers [16, 20, 21, 25, 31, 41, 43, 44], and seven of them are considered to be reproducible methods in our analysis [16, 20, 21, 25, 31, 43, 44].[6] Of the seven reproducible methods, four methods [16, 21, 31, 44] were published in 2020 and 20201, and have not been compared with each other.

As simple yet effective baselines, we include two widely known frequency-based methods—*global top-frequency* (G-TopFreq) and *personal top-frequency* (P-TopFreq), which are often shown as simple baselines in recommendation tasks. Surprisingly, we find that three of the five deep learning based methods that we consider only compare with the global top-frequency baseline G-TopFreq [25, 31, 43], but do not compare to the personal top-frequency baseline P-TopFreq, which is known to have higher performance in general [16, 20, 21, 44].

There is an important limitation of P-TopFreq w.r.t. the basket size in an NBR setting that is ignored in previous work. P-TopFreq can only recommend items from the past transactions of a user, which means that it might not be able to fully make use of the available basket slots like other methods, and this may lead to an unfair comparison. We analyze the percentage of basket slots used for P-TopFreq, and Table 2 shows the results. To address this limitation, we propose a simple combination of G-TopFreq and P-TopFreq as an additional baseline, called *GP-TopFreq*: GP-TopFreq first uses P-TopFreq to fill a basket, then uses G-TopFreq to fill any remaining slots.

3.2.1 Frequency-Based Baselines.

- *G-TopFreq* uses the k most popular items in the dataset to form the recommended next basket. It is widely used in recommendation systems due to its effectiveness and simplicity.
- *P-TopFreq* is a personalized top frequency method that treats the most frequent k items in the users' historical records as the next basket. This method only has repeat items in the prediction.
- *GP-TopFreq* is a combination of P-TopFreq and G-TopFreq to make full use of the available basket slots.

3.2.2 Nearest Neighbor Based Methods.

- *TIFUKNN* models the temporal dynamics of frequency information of users' past baskets to introduce personalized frequency information, then uses a nearest neighbor based method on personalized frequency information [21].

---

[4]We first check whether the paper provides a link to their code; if not, we search GitHub using the title of the paper.
[5]We re-implement the Dream algorithm [43].
[6]We do not include FPMC [33] in our work for the following two reasons: first, it will break the selection criteria [15] we employ in this article, and second, FPMC is among the worst-performing methods in all NBR related papers [21, 25, 31, 43] recently.

- *UP-CF@r* is a combination of recency-aware, user-wise popularity and user-wise collaborative filtering. The recency of shopping behavior is considered in this method [16].

3.2.3 Deep Learning Based Methods.

- *Dream* is the first deep learning based method that models users' global sequential basket history for NBR. It uses a pooling strategy to generate basket representations, which are then fed into an RNN to learn user representations and predict the corresponding next set of items [43].
- *Sets2Sets* uses a pooling operation to get basket embeddings and an attention mechanism to learn a user's representation from their past interactions. Furthermore, item frequency information is adopted to improve performance [20].
- *DNNTSP* leverages a GNN and self-attention techniques. It encodes item-item relations via a graph and employs a self-attention mechanism to capture temporal dependencies of users' basket sequence [44].
- *Beacon* is an RNN-based method that encodes the basket considering the incorporating information on pairwise correlations among items [25].
- *CLEA* is an RNN-based method that uses a contrastive learning model to automatically extract items relevant to the target items and generates the representation via a GRU-based encoder [31].

3.3 Implementation Details

For deep learning based methods [20, 25, 31, 43, 44], we strictly follow the hyper-parameter setting and tuning strategy in their respective paper or related GitHub repository. Following the same strategy as Dacrema et al. [15], we use the suggested best parameters in TIFUKNN [21] to achieve its best performance. For UP-CF@r, the recency window is tuned on {1, 5, 10, 50}, the locality is tuned on {1, 20, 50, 100}, and the asymmetry is tuned on {0, 0.25, 0.5, 0.75, 1.0}. We perform a grid search on the validation dataset to tune hyper-parameters and select the best model for testing. For all methods, we rely as much as possible on the original source code and construct a pipeline to perform experiments. We share the code and data used in our experiments online.[7]

4 PERFORMANCE COMPARISON USING CONVENTIONAL METRICS

4.1 Conventional NBR Metrics

To analyze the performance of NBR methods, we first consider three conventional metrics: recall, Normalized Discounted Cumulative Gain (NDCG) and Personalized Hit Ratio (PHR), all of which are commonly used in previous NBR studies [20, 21, 44]. We do not consider the F1 and Precision metrics in this article, since we focus on the basket recommendation with a fixed basket size K, which means the Precision@K and F1@K are proportional to Recall@K for each user. F1@K and Precision@K are more suitable for NBR with a dynamic basket size for each user. Notations used throughout this work are presented in Table 3.

Recall measures the ability to find all relevant items and is calculated as follows:

$$Recall@K(u_j) = \frac{\left|P_{u_j} \cap T_{u_j}\right|}{\left|T_{u_j}\right|}, \tag{1}$$

where P_{u_j} is the predicted basket with K recommended items and T_{u_j} is the ground-truth basket for user u_j. The average recall score of all users is adopted as the recall performance.

---

[7]Code available at https://github.com/liming-7/A-Next-Basket-Recommendation-Reality-Check.

Table 3. Notation Used in the Article

Symbol	Description
U	Set of all users
U_g	Set of users in group g
I	Set of all items
u	A single user in U
i	A single item in I
S_j	Sequence of historical baskets for u_j
B_j^t	The t-th basket in S_j, a set of items $i \in I$
T_{u_j}	Target/ground-truth basket for u_j that we aim to predict
$T_{u_j}^{rep}$	Set of *repeat items* in the ground-truth basket T_{u_j} for u_j
$T_{u_j}^{expl}$	Set of *explore items* in the ground-truth basket T_{u_j} for u_j
P_{u_j}	Predicted basket for u_j
$P_{u_j}^{rep}$	Set of *repeat items* in the predicted basket P_{u_j} for u_j
$P_{u_j}^{expl}$	Set of *explore items* in the predicted basket P_{u_j} for u_j
$I_{j,t}^{rep}$	Repeat items for u_j at timestamp t; set of items that have appeared in u_j's baskets up to timestamp t
$I_{j,t}^{expl}$	Explore items for u_j at timestamp t; set of items that have not appeared in u_j's baskets up to timestamp t

NDCG is a ranking quality measurement metric, which takes item order into consideration and it is calculated as follows, for a user $u \in U$ and its ground-truth basket T_u:

$$\text{NDCG@}K(u_j) = \frac{\sum_{k=1}^{K} p_k / \log_2(k+1)}{\sum_{k=1}^{\min(K, |T_{u_j}|)} 1 / \log_2(k+1)}, \tag{2}$$

where p_k equals 1 if $P_{u_j}^k \in T_u$, and otherwise $p_k = 0$. P_u^k denotes the k-th item in the predicted basket P_u. The average score across all users is the NDCG performance of the algorithm.

PHR focuses on user-level performance and calculates the ratio of predictions that capture at least one item in the ground-truth basket as follows:

$$\text{PHR@}K = \frac{\sum_{j=1}^{N} \varphi(P_{u_j}, T_{u_j})}{N}, \tag{3}$$

where N is the number of test users, and $\varphi(P_{u_j}, T_{u_j})$ returns 1 when $P_{u_j} \cap T_{u_j} \neq \emptyset$ and otherwise returns 0.

4.2 Results with Conventional NBR Metrics

Performance results for the conventional NBR metrics are shown in Table 4. The performance of different methods varies across datasets; there is no method that consistently outperforms all other methods, independent of dataset and basket size. This calls for a further analysis of the factors impacting performance, which we conduct in the next section.

Among the frequency-based baselines, P-TopFreq outperforms G-TopFreq in all scenarios, which indicates that personalization improves the NBR performance. P-TopFreq can only recommend items that have appeared in a user's previous baskets. As pointed out in Section 3.2, the number of repeat items of a user may be smaller than the basket size, which means there might be empty slots in a basket recommended by P-TopFreq. Despite this limitation, P-TopFreq is a competitive

Table 4. Overall Performance Comparison of Frequency-Based, Nearest Neighbor Based, and Deep Learning Based NBR Methods

Size		10			20		
Dataset	Methods	Recall	NDCG	PHR	Recall	NDCG	PHR
TaFeng	G-TopFreq	0.0831 (0.0018)	0.0864 (0.0017)	0.2498 (0.0024)	0.1114 (0.0029)	0.0961 (0.0018)	0.3311 (0.0006)
	P-TopFreq	0.1069 (0.0023)	0.0955 (0.0019)	0.3473 (0.0033)	0.1395 (0.0026)	0.1096 (0.0019)	0.4329 (0.0038)
	GP-TopFreq	0.1211 (0.0031)	0.1015 (0.0023)	0.3691 (0.0043)	0.1693 (0.0031)	0.1208 (0.0022)	0.4834 (0.0040)
	UP-CF@r	0.1249 (0.0027)	0.1104 (0.0019)	0.3983 (0.0035)	0.1694 (0.0034)	0.1280 (0.0021)	0.4877 (0.0048)
	TIFUKNN	0.1251 (0.0033)	0.1016 (0.0014)	0.3852 (0.0029)	0.1817 (0.0037)	0.1232 (0.0016)	0.5043 (0.0035)
	Dream	0.1134 (0.0023)	0.1022 (0.0018)	0.3035 (0.0024)	0.1463 (0.0034)	0.1149 (0.0021)	0.3905 (0.0039)
	Beacon	0.1139 (0.0032)	0.1033 (0.0023)	0.3055 (0.0044)	0.1475 (0.0026)	0.1154 (0.0020)	0.4002 (0.0024)
	CLEA	0.1184 (0.0038)	0.1046 (0.0030)	0.3076 (0.0044)	0.1477 (0.0035)	0.1165 (0.0028)	0.3957 (0.0037)
	Sets2Sets	0.1349 (0.0034)	0.1124 (0.0025)	0.4080 (0.0058)	0.1904 (0.0021)	0.1342 (0.0020)	0.5261 (0.0041)
	DNNTSP	0.1526 (0.0034)*	0.1314 (0.0022)*	0.4460 (0.0044)*	0.2077 (0.0044)*	0.1532 (0.0025)*	0.5496 (0.0038)*
Dunnhumby	G-TopFreq	0.0982 (0.0009)	0.1050 (0.0010)	0.4621 (0.0033)	0.1264 (0.0006)	0.1192 (0.0008)	0.5314 (0.0030)
	P-TopFreq	0.2319 (0.0013)	0.2340 (0.0012)	0.6559 (0.0017)	0.3030 (0.0012)	0.2695 (0.0011)	0.7173 (0.0017)
	GP-TopFreq	0.2356 (0.0013)	0.2358 (0.0013)	0.6649 (0.0019)	0.3141 (0.0012)	0.2743 (0.0011)	0.7372 (0.0026)
	UP-CF@r	0.2428 (0.0006)	0.2468 (0.0012)	0.6747 (0.0024)	0.3185 (0.0009)	0.2848 (0.0013)	0.7339 (0.0014)
	TIFUKNN	0.2396 (0.0008)	0.2408 (0.0011)	0.6761 (0.0022)	0.3191 (0.0015)	0.2799 (0.0012)	0.7407 (0.0030)
	Dream	0.0950 (0.0008)	0.1036 (0.0008)	0.4586 (0.0040)	0.1300 (0.0013)	0.1205 (0.0011)	0.5395 (0.0033)
	Beacon	0.0995 (0.0009)	0.1066 (0.0011)	0.4700 (0.0037)	0.1354 (0.0009)	0.1241 (0.0010)	0.5506 (0.0025)
	CLEA	0.1552 (0.0010)	0.1732 (0.0010)	0.5541 (0.0038)	0.1866 (0.0012)	0.1864 (0.0007)	0.6273 (0.0029)
	Sets2Sets	0.1691 (0.0023)	0.1473 (0.0015)	0.5802 (0.0053)	0.2552 (0.0018)	0.1880 (0.0015)	0.6893 (0.0046)
	DNNTSP	0.2404 (0.0007)	0.2430 (0.0010)	0.6767 (0.0022)	0.3242 (0.0004)*	0.2839 (0.0007)	0.7427 (0.0016)
Instacart	G-TopFreq	0.0710 (0.0003)	0.0811 (0.0001)	0.4542 (0.0010)	0.0990 (0.0001)	0.0962 (0.0002)	0.5248 (0.0017)
	P-TopFreq	0.3260 (0.0008)	0.3378 (0.0007)	0.8449 (0.0018)	0.4307 (0.0008)	0.3939 (0.0002)	0.8957 (0.0019)
	GP-TopFreq	0.3269 (0.0008)	0.3383 (0.0007)	0.8463 (0.0018)	0.4354 (0.0007)	0.3961 (0.0003)	0.9011 (0.0017)
	UP-CF@r	0.3506 (0.0007)	0.3631 (0.0007)	0.8652 (0.0020)	0.4591 (0.0008)	0.4222 (0.0005)	0.9079 (0.0012)
	TIFUKNN	0.3601 (0.0015)*	0.3721 (0.0008)*	0.8642 (0.0005)	0.4709 (0.0015)*	0.4323 (0.0003)*	0.9097 (0.0011)
	Dream	0.0712 (0.0004)	0.0805 (0.0005)	0.4551 (0.0008)	0.0997 (0.0006)	0.0957 (0.0007)	0.5304 (0.0027)
	Beacon	0.0734 (0.0003)	0.0838 (0.0003)	0.4628 (0.0006)	0.1050 (0.0006)	0.1009 (0.0004)	0.5462 (0.0022)
	CLEA	0.1221 (0.0014)	0.1449 (0.0016)	0.5603 (0.0053)	0.1514 (0.0045)	0.1592 (0.0019)	0.6347 (0.0094)
	Sets2Sets	0.2125 (0.0013)	0.1923 (0.0019)	0.7185 (0.0040)	0.3077 (0.0040)	0.2402 (0.0020)	0.8284 (0.0040)
	DNNTSP	0.3330 (0.0003)	0.3412 (0.0004)	0.8525 (0.0011)	0.4423 (0.0005)	0.4000 (0.0005)	0.9042 (0.0003)

Highlights indicate the highest score per basket size and metric. An asterisk (*) indicates that the highest score for a given basket size and metric is significantly better than the second highest score (paired t-test, p-value < 0.05).

NBR baseline. GP-TopFreq makes full use of the available basket slots by filling any slots with top-ranked items suggested by G-TopFreq. GP-TopFreq outperforms P-TopFreq with no surprise, and, as expected, the difference shrinks as the repetition ratio of the dataset increases. For future fair comparisons, we believe that GP-TopFreq should be the baseline for every NBR method to compare with, especially in high exploration scenarios, to be able to determine what value is added beyond simple frequency-based recommendations.

As to the nearest neighbor based methods, we see that TIFUKNN and UP-CF@r have a similar performance across different scenarios and outperform all frequency-based baselines. The two methods are similar in the sense that both model temporal information, combined with a user-based nearest neighbor method. Their performance in a high exploration scenario is lower than several deep learning based methods (i.e., the TaFeng dataset), but on the Dunnhumby and Instacart datasets, which have a relatively low exploration ratio, they are among the best-performing methods.

Most of the deep learning based methods outperform G-TopFreq, which is the only frequency-based baseline considered in many papers. Surprisingly, P-TopFreq and GP-TopFreq achieve a highly competitive performance and outperform four deep learning based methods (i.e., Dream, Beacon, CLEA, and Sets2Sets) by a large margin in the Dunnhumby and Instacart datasets, where the improvements in terms of *Recall*@10 range from 35.8% to 141.9% and from 53.6% to 353.3%, respectively. Moreover, the proposed GP-TopFreq baseline outperforms the deep learning based Beacon, Dream, and CLEA algorithms on the TaFeng dataset, the scenario with a high exploration ratio. Of the deep learning based methods, DNNTSP is the only one to have consistently high performance in all scenarios.

4.3 Upshot

Based on the preceding experiments and analysis, we conclude that the choice of dataset plays an important role in evaluating the performance of NBR methods, and no state-of-the-art NBR method is able to consistently achieve the best performance across datasets.

Several deep learning based NBR methods [25, 31, 43] aim to learn better performing representations by capturing long-term temporal dependencies, denoising, and so forth. They do indeed outperform the G-TopFreq baseline, but many are inferior to the P-TopFreq baseline, especially in datasets with a relatively high repetition ratio. The proposed GP-TopFreq baseline in some sense "re-calibrates" the improvements reported for recently introduced complex, deep learning based NBR methods; compared to the simple GP-TopFreq baseline, their improvements are modest or even absent.[8]

So far, we have used conventional metrics to examine the performance of NBR methods. To account for the findings reported in this section and provide insights for future NBR method development, we will now consider additional metrics.

5 PERFORMANCE W.R.T. REPETITION AND EXPLORATION

To understand which factors contribute to the overall performance of an NBR method, we dive deeper into the basket components from a repetition and exploration perspective.

5.1 Metrics for Repetition vs. Exploration

We propose several metrics that are meant to capture repetition and exploration aspects of a basket. First, we adopt widely used definitions of repetition and exploration in the recommender systems literature [2, 10, 11, 36] to define what constitutes a repeat item and an explore item in the context of NBR. Specifically, an item i^r is considered to be a *repeat item* for a user u_j if it appears in the sequence of the user's historical baskets S_j—that is, if $i^r \in I_{j,t}^{rep} = B_j^1 \cup B_j^2 \cup \cdots \cup B_j^t$. Otherwise, the item is an *explore item*, denoted as $i^e \in I_{j,t}^{expl} = I - I_{j,t}^{rep}$. We write $P_{u_j} = P_{u_j}^{rep} \cup P_{u_j}^{expl}$ for the predicted next basket B_j^{t+1}, which is the union of *repeat items* $P_{u_j}^{rep}$ and *explore items* $P_{u_j}^{expl}$. As an edge case, a basket may consist of repeat or explore items only.

Novelty of recommendation is a concept that is similar to, but different from, the notion of exploration that we use in this work. Several novelty-related metrics have been proposed (i.e., EPC and EPD [36]). It is important to note that these metrics are not suitable for our analysis in this article. First, they only focus on measuring the novelty of a ranked list, whereas we want to not only understand the components within the predicted basket but also analyze a model's ability to fulfill a user's needs w.r.t. repetition and exploration. Second, these metrics are not NBR specific and only

---

focus on one aspect (i.e., novelty), whereas we want to make a comparison between repetition and exploration to assess the NBR performance.

To analyze the composition of a predicted basket, we propose the *repetition ratio*, *RepR*, and the *exploration ratio*, *ExplR*. *RepR* and *ExplR* represent the proportion of *repeat items* and *explore items* in a recommended basket, respectively. The overall *RepR* and *ExplR* scores are calculated over all test users as follows:[9]

$$RepR = \frac{1}{N} \sum_{j=1}^{N} \frac{\left|P_{u_j}^{rep}\right|}{K}, \qquad ExplR = \frac{1}{N} \sum_{j=1}^{N} \frac{\left|P_{u_j}^{expl}\right|}{K}, \tag{4}$$

where N denotes the number of test users; K is the size of the model's predicted basket for user u_j; and $P_{u_j}^{rep}$ and $P_{u_j}^{expl}$ are the sets of repeat items in P_{u_j} and of explore items in P_{u_j}, respectively.

Next, we pay attention to a basket's ability to fulfill a user's need for repetition and exploration, and propose the following metrics. $Recall_{rep}$ and PHR_{rep} are used to evaluate the *Recall* and *PHR* w.r.t. the repetition performance; similarly, we use $Recall_{expl}$ and PHR_{expl} to assess the exploration performance. More precisely:

$$Recall_{rep} = \frac{1}{N_r} \sum_{j=1}^{N_r} \frac{\left|P_{u_j} \cap T_{u_j}^{rep}\right|}{\left|T_{u_j}^{rep}\right|}, \qquad PHR_{rep} = \frac{\sum_{j=1}^{N_r} \varphi\left(P_{u_j}, T_{u_j}^{rep}\right)}{N_r} \tag{5}$$

and

$$Recall_{expl} = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{\left|P_{u_j} \cap T_{u_j}^{expl}\right|}{\left|T_{u_j}^{expl}\right|}, \qquad PHR_{expl} = \frac{\sum_{j=1}^{N_e} \varphi\left(P_{u_j}, T_{u_j}^{expl}\right)}{N_e}, \tag{6}$$

where N_r and N_e denote the number of users whose ground-truth basket contains *repeat items* and *explore items*, respectively; $\varphi(P, T)$ returns 1 when $P \cap T \neq \emptyset$ and otherwise returns 0.

Next, we first use the repetition ratio and exploration ratio to examine the recommended baskets; we then use our repetition and exploration metrics to re-assess the NBR methods that we consider, examine how repetition and exploration contribute to the overall recommendation performance, and how users with different degrees of repeat behavior benefit from different NBR methods.

5.2 The Components of a Recommended Basket

We analyze the components of the recommended basket for each NBR method to understand what makes up the recommendation. The results are shown in Figure 2. First, we see that, averaged over all users, all recommended baskets are heavily skewed toward either item repetition or exploration, relative to the ground-truth baskets that are much more balanced between already seen and new items. We can divide the methods that we compare into repeat-biased methods (i.e., P-TopFreq, GP-TopFreq, Sets2Sets, DNNTSP, UP-CF@r, and TIFUKNN) and explore-biased methods (i.e., G-TopFreq, Dream, Beacon, and CLEA). Importantly, a large performance gap exists between the two types. None of the published NBR methods can properly balance the repeat items and explore items of users' future interests. Figure 3 shows the repetition ratio *RepR* distribution for the ground-truth basket and the recommended basket derived by a repeat-biased method or an explore-biased method. We show the *RepR* distribution of a representative explore-biased method (i.e., CLEA) and a representative repeat-biased method (i.e., DNNTSP) in Figure 3. The *RepR* distribution of the other eight NBR methods are provided in the appendix (see Figure 7).

---

[9]To assess this performance on a dataset, we use the average performance across users; we also show the corresponding user-level *RepR* distribution in our analysis.
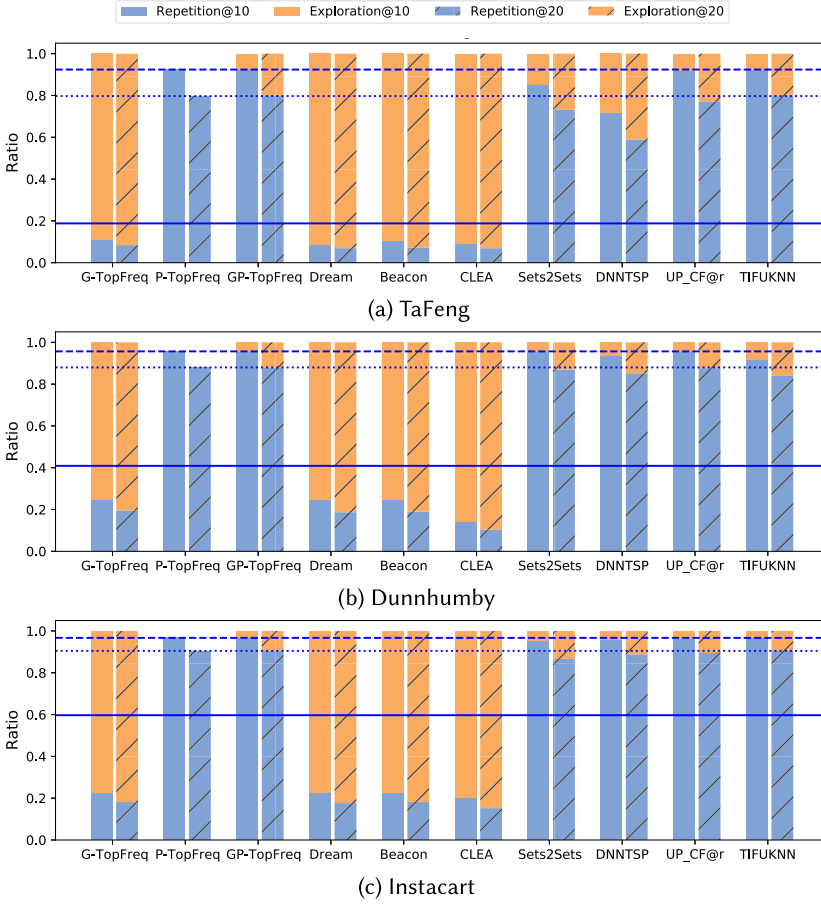
(a) TaFeng

(b) Dunnhumby

(c) Instacart

Fig. 2. The repetition ratio *RepR* and exploration ratio *ExplR* of recommended baskets averaged over all users. Solid lines indicate the repetition ratio of the ground truth, and dashed lines indicate the upper bound of the repetition ratio for basket size 10 and dotted lines for basket size 20.

Among the explore-biased methods, G-TopFreq is not a personalized method; it always provides the most popular items. Dream, Beacon, and CLEA treat all items without any discrimination, which means the explore items are more likely to be in the predicted basket and their basket components are similar to G-TopFreq. Looking at the performance in Table 4, we see that repeat-biased methods generally perform much better than explore-biased methods on conventional metrics across the datasets, especially when the dataset has a relatively high repetition ratio.

The repetition ratios of P-TopFreq and GP-TopFreq serve as the upper bound repetition ratio for the recommended basket. Most baskets recommended by repeat-biased methods are close to or reach this upper bound, even when the datasets have a low ratio of repeat behavior in the ground truth, except for two cases (Sets2Sets and DNNTSP on the TaFeng dataset).

Finally, the exploration ratio of repeat-biased methods increases from basket size 10 to 20; we believe that this is because there are simply no extra repeat items available: it does not mean that the methods actively increase the exploration ratio in a larger basket setting.
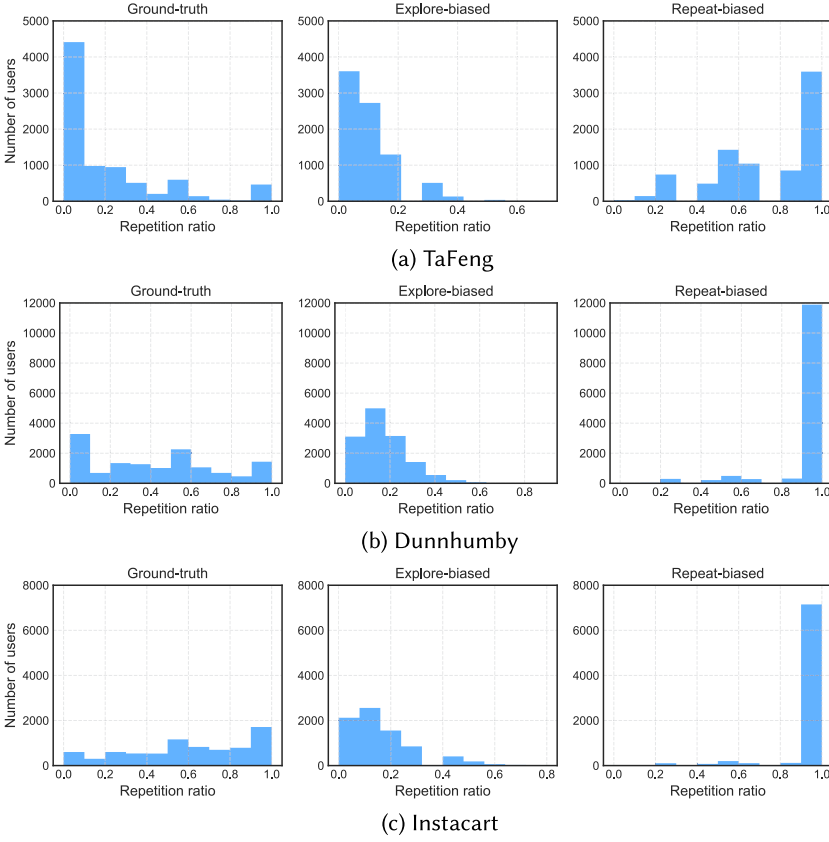
Fig. 3. Distribution of the repetition ratio *RepR* of recommended baskets on different datasets for an explore-biased method (CLEA) and a repeat-biased method (DNNTSP).

5.3 Performance w.r.t. Repetition and Exploration

The results in terms of repetition and exploration performance are shown in Table 5. First of all, using our proposed metrics, we observe that the repetition performance $Recall_{rep}$ is always higher than the exploration performance, even when the explore items form almost 90% of the recommended basket. This shows that the repetition task (recommending repeat items) and the exploration task (recommending explore items) have different levels of difficulty and that capturing users' repeat behavior is much easier than capturing their explore behavior.

Three deep learning based methods perform worst w.r.t. repeat item recommendation *and* best w.r.t. explore item recommendation at the same time, as they are heavily skewed toward explore items. We also see that there are improvements in the exploration performance compared to G-TopFreq with the same level of exploration ratio, which indicates that the representation learned by these methods does capture the hidden sequential transition relationship between items. Repeat-biased methods perform better w.r.t. repetition in all settings, since the baskets they predict contain more repeat items. Similarly, we can see that DNNTSP, UP-CF@r, and TIFUKNN perform better than P-TopFreq w.r.t. repeat performance with the same or a lower level of repetition ratio.

Third, explore-biased methods spend more resources on the more difficult and uncertain task of explore item prediction, which is not an optimal choice when considering the overall NBR

Table 5. Repetition and Exploration Performance Comparison of Frequency-Based, Nearest Neighbor Based, and Deep Learning Based NBR Methods

Size	Dataset Methods	TaFeng				Dunnhumby				Instacart			
		Recall -rep	Recall -expl	PHR -rep	PHR -expl	Recall -rep	Recall -expl	PHR -rep	PHR -expl	Recall -rep	Recall -expl	PHR -rep	PHR -expl
10	G-TopFreq	0.1268	0.0573	0.1947	0.1738	0.1882	0.0393	0.4954	0.1590	0.1002	0.0382	0.4138	0.1575
	P-TopFreq	0.5234	0.0000	0.6766	0.0000	0.5612	0.0000	0.8553	0.000	0.5388	0.0000	0.9085	0.0000
	GP-TopFreq	0.5234	0.0157	0.6766	0.0266	0.5612	0.0049	0.8553	0.0148	0.5388	0.0014	0.9085	0.0046
	UP-CF@r	0.6046*	0.0086	0.7515*	0.0153	0.5913*	0.0018	0.8752*	0.0064	0.5805	0.0011	0.9295	0.0038
	TIFUKNN	0.5616	0.0176	0.7037	0.0284	0.5726	0.0082	0.8646	0.0257	0.5931*	0.0018	0.9287	0.0060
	Dream	0.1312	0.0921	0.1874	0.2378	0.1843	0.0412	0.4906	0.1668	0.1007	0.0367	0.4181	0.1503
	Beacon	0.1456	0.0822	0.2126	0.2223	0.1893	0.0404	0.4963	0.1605	0.1010	0.0365	0.4193	0.1492
	CLEA	0.1442	0.0935	0.2073	0.2398	0.2492	0.0569*	0.5702	0.1803*	0.1715	0.0337	0.5435	0.1297
	Sets2Sets	0.5647	0.0271	0.7222	0.0495	0.4260	0.0026	0.7583	0.0064	0.3515	0.0005	0.7713	0.0013
	DNNTSP	0.5291	0.0556	0.6912	0.1294	0.5861	0.0073	0.8684	0.0224	0.5477	0.0018	0.9110	0.0054
20	G-TopFreq	0.1637	0.0789	0.2530	0.2385	0.2279	0.0609	0.5565	0.2353	0.1335	0.0602	0.4767	0.2251
	P-TopFreq	0.7251	0.0000	0.8439	0.0000	0.7399	0.0000	0.9353	0.0000	0.7193	0.0000	0.9631	0.0000
	GP-TopFreq	0.7251	0.0339	0.8439	0.0724	0.7399	0.0153	0.9353	0.0470	0.7193	0.0084	0.9631	0.0285
	UP-CF@r	0.7785*	0.0259	0.8742	0.0560	0.7718	0.0078	0.9430*	0.0261	0.7612	0.0083	0.9735	0.0215
	TIFUKNN	0.7664	0.0410	0.8707	0.0806	0.7474	0.0159	0.9344	0.0677	0.7747*	0.0108	0.9728	0.0318
	Dream	0.1723	0.1234	0.2572	0.3264	0.2316	0.0687	0.5612	0.2596	0.1348	0.0586	0.4849	0.2222
	Beacon	0.1748	0.1230	0.2621	0.3262	0.2333	0.0672	0.5628	0.2568	0.1338	0.0599	0.4822	0.2252
	CLEA	0.1799	0.1217	0.2675	0.3189	0.2808	0.0874*	0.6168	0.2648*	0.2081	0.0578	0.6051	0.2044
	Sets2Sets	0.7478	0.0558	0.8703	0.1166	0.6317	0.0086	0.8881	0.0273	0.5095	0.0032	0.8873	0.0073
	DNNTSP	0.6820	0.0899	0.8130	0.2103	0.7758	0.0183	0.9391	0.0609	0.7284	0.0095	0.9657	0.0320

Highlights indicate the highest score per basket size, for the exploration and repetition metrics. As in Table 4, an asterisk (*) indicates that the highest score for a given basket size and metric is significantly better than the second highest score (paired t-test, p-value < 0.05).

performance. Being biased toward the easier task of repeat item prediction leads to gains in the overall performance, which is positively correlated with the repetition ratio of the dataset.

To understand the potential reasons for a method being repeat biased or explore biased, we provide an in-depth analysis of the methods' architectures. P-TopFreq and GP-TopFreq are repeat-biased methods, as they both mainly rely on the frequency of historical items to recommend the next basket. Two nearest neighbor based methods (i.e., TIFUKNN and UP-CF@r) have a module to model both the frequency and the recency of historical items; besides, they both have a parameter to emphasize the frequency and recency information. Similarly, Sets2Sets is also repeat biased, as it adds the historical items' frequency information to the prediction layer. DNNTSP does not consider frequency information; however, it has an indicator vector to indicate whether an item has appeared in the historical basket sequence or not, which can be regarded as a repeat item indicator. G-TopFreq is explore biased since it is not a personalized method and can only recommend top-K popular items within the dataset. The remaining three explore-biased methods (Dream, Beacon, and CLEA) do not consider the frequency of historical items or the indicator of items' appearance, so they fail to identify the benefits of recommending repeat items.

5.4 The Relative Contribution of Repetition and Exploration

Even though a clear improvement w.r.t. either repeat or explore performance can be observed in the previous section, this does not mean that this improvement is the reason for the better overall performance, since repeat and explore items account for different ground-truth proportions in different datasets. To better understand where the performance gains of the well-performing methods in Table 4 come from, we remove *explore items* and keep *repeat items* in the predicted basket to compute the contribution of repetition, and similarly, we remove *repeat items* and keep *explore items* to compute the performance, which can be regarded as the contribution of exploration.

Experimental results on three datasets are shown in Figure 4. We consider G-TopFreq, P-TopFreq, and GP-TopFreq as simple baselines to compare with. From Figure 4, we conclude that Dream and Beacon perform better than G-TopFreq on the TaFeng dataset, as the main performance gain is from improvements in the exploration prediction. As a consequence, in the Dunnhumby and Instacart datasets, Dream, Beacon, and G-TopFreq achieve similar performance, and the repeat prediction contributes the most to the overall performance, even when their recommended items are heavily skewed toward explore items. Additionally, we observe that CLEA outperforms other explore-biased methods due to its improvements in the repetition performance without sacrificing the exploration performance.

At the same time, it is clear that TIFUKNN, UP-CF@r, Sets2Sets, and DNNTSP outperform explore-biased methods because of the improvements in the repetition performance, even at the detriment of exploration. The repeat items make up the majority of their correct recommendations. Specifically, repeat recommendations contribute to more than 97% of their overall performance on the Dunnhumby and Instacart datasets.

An interesting comparison is between Sets2Sets and P-TopFreq. The strong performance gain of Sets2Sets on the TaFeng dataset is mainly due to the exploration part, whereas P-TopFreq outperforms it by a large margin on the other two datasets at the same level of repetition ratio, even though the personal frequency information is considered in the Sets2Sets model. We believe this indicates that the loss on repeat items seems to be suppressed by the loss on explore items during the training process, which weakens the influence of the frequency information.

Recall that the number of repetition candidates for a user may be smaller than the basket size, which means that there might be empty slots in the basket recommended by P-TopFreq. From Figure 2 and Table 2, we observe that the empty slots account for a significant proportion of exploration slots in many settings. However, existing studies omit this fact when making the
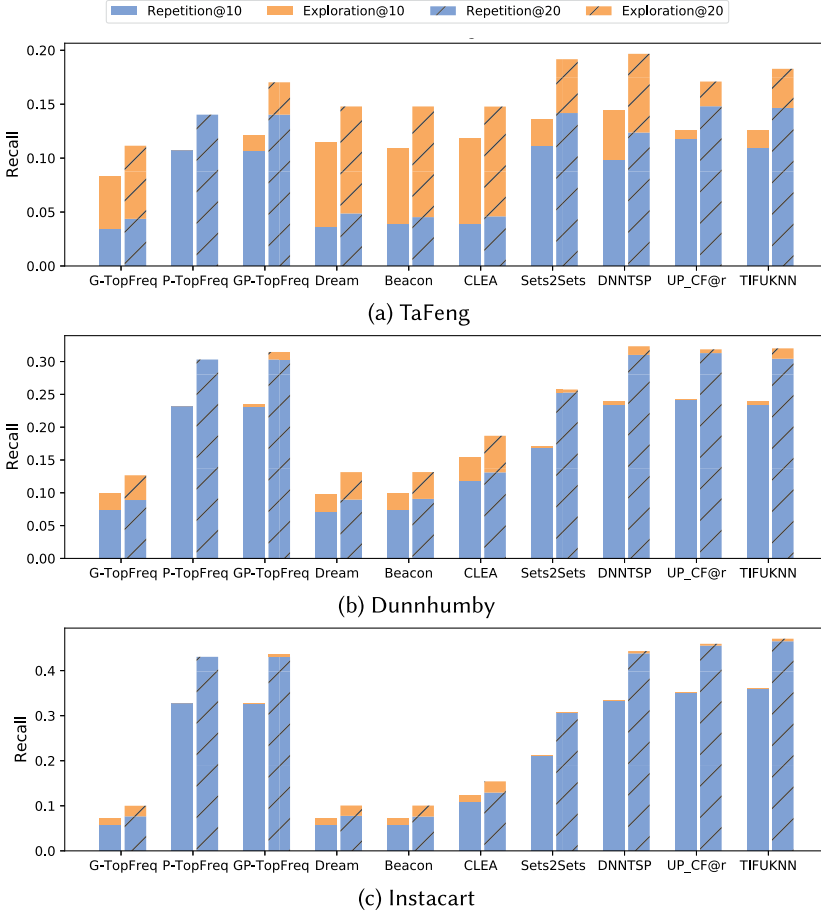
Fig. 4. Performance contribution from repeat and explore recommendations on the TaFeng, Dunnhumby, and Instacart datasets.

comparison with P-TopFreq, leading to an unfair comparison and overestimation of the improvement, as their predictions leverage more slots. For example, Dream, Beacon, and CLEA can beat P-TopFreq, but they are inferior to GP-TopFreq. TIFUKNN and UP-CF@r model the temporal order of the frequency information, leading to a higher repetition performance than P-TopFreq in general. Even though the contribution of the repetition performance improvement is obvious on the Instacart dataset, it is less meaningful on the other two datasets, where the performance gain is mainly from the exploration part by filling the empty slots. When compared with the proposed GP-TopFreq baseline on the TaFeng and Dunnhumby datasets, the improvement is around a modest 3%.

DNNTSP is always among the best-performing methods across the three datasets and is able to model exploration more effectively than other repeat-biased methods. Moreover, it also actively recommends explore items rather than being totally biased toward the repeat recommendation in high exploration scenarios. However, the improvement is limited due to the relatively high repetition ratios and the huge difficulty gap between repetition and exploration tasks. Compared with GP-TopFreq, the improvement of DNNTSP w.r.t. *Recall*@10 on the Dunnhumby and Instacart

datasets is merely 1.3% and 1.9%, respectively, which is modest considering the complexity and training time added by DNNTSP.

Obviously, even though many advanced NBR algorithms learn rich user and/or item representations, the main performance gains stem from the prediction of repeat behavior. Yet, limited progress w.r.t. overall performance has so far been made compared to the simple P-TopFreq and GP-TopFreq baseline methods.

5.5 Treatment Effect for Users with Different Repetition Ratios

As the average repetition ratio in a dataset has a significant influence on a model's performance (see Section 4), existing NBR methods are skewed to repetition or exploration (see Section 5.2), and global trend might influence the users' repetition patterns, it is of interest to investigate the treatment effect for users with different repetition ratios. We examine the performance of NBR methods w.r.t. different groups of users with different repetition ratios. We divide the users into five groups according to their repetition ratio ([0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1]) and calculate the average performance within each group. Note that the repetition ratio indicates the user's preference w.r.t. *repeat items* and *explore items* (e.g., users with a low repetition ratio prefer to purchase new items in their next basket). The results are shown in Figure 5.

First, we can see that the methods' performance within different user groups is different from the performance computed over all users (see Table 4). For example, several explore-biased methods (G-TopFreq, Dream, Beacon, CLEA) can outperform recent repeat-biased methods (TIFUKNN, UP-CF@r, Sets2Sets, DNNTSP) in the user group with a low repetition ratio ([0, 0.2]), but these explore-biased methods are inferior to the repeat-biased methods when computing the performance over all users. Second, the performance of repeat-biased NBR models increases as the repetition ratio increases. Interestingly, we observe an analogous trend w.r.t. the performance of explore-biased NBR methods as the repetition ratio increases, but the rate of the increase is smaller. We believe that this is because the NBR task gets easier for users with a higher repetition ratio, and the repeat-biased methods benefit more from an increase in repetition ratio.

From the perspective of user group fairness, explore-biased methods seem to be fairer than repeat-biased methods across different user groups, as they have quite similar performance across groups. Explore-biased methods have lower variation in performance than repeat-biased methods. However, we should be aware of intrinsic difficulty gaps between different user groups (e.g., it is easier for NBR methods to find correct items for users who like to repeat a purchase). Taking this into consideration, we take G-TopFreq and GP-TopFreq as two anchor baselines to evaluate whether recent NBR methods put a specific user group at a disadvantage or not. On the TaFeng and Dunnhumby datasets, repeat-biased methods (Sets2Sets, UP-CF, TIFUKNN, DNNTSP) fail to achieve the performance of G-TopFreq within users whose repetition ratio is in [0, 0.2], which means they do not cater to users of this group. At the same time, recent explore-biased methods (Dream, Beacon, CLEA) fail to achieve the performance derived by the very simple baseline (i.e., GP-TopFreq) on four user groups on TaFeng, Dunnhumby, and Instacart datasets. This analysis indicates that both repeat-biased and explore-biased NBR methods do not treat all user groups fairly.

5.6 Looking Beyond the Average Performance

In the recommender systems literature, it is customary to compute the average performance over all test users to represent the performance of a recommendation method. Given the diverse treatment effect across different user groups, we want to drill down and see how much the different user groups contribute toward the overall average performance. As before, we use five groups as defined in Section 5.5 in terms of the repetition ratio. Specifically, for each individual group g_j, we

(a) TaFeng
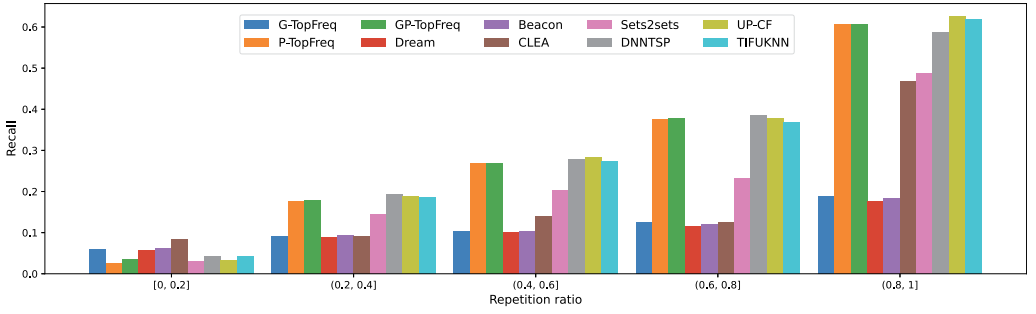
(b) Dunnhumby
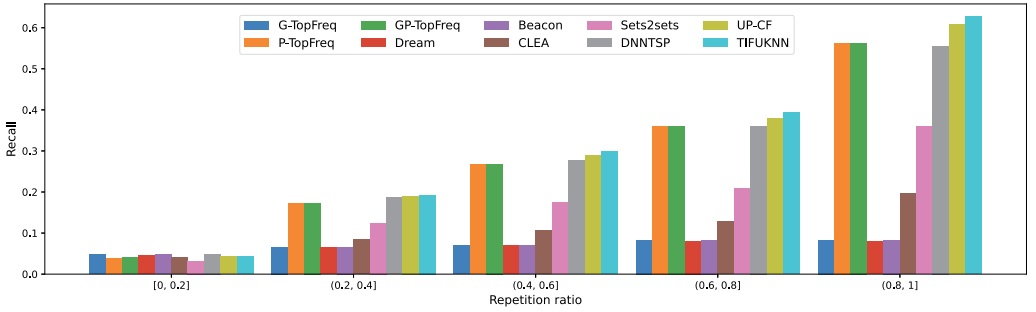
(c) Instacart

Fig. 5. Treatment effect for users on the TaFeng, Dunnhumby, and Instacart datasets for 10 NBR methods for users with different repetition ratios (binned in five groups).

analyze its *proportion of all users* (*PAU*) and its *contribution to the average performance* (*CAP*) as follows:

$$PAU_j = \frac{|U_{g_j}|}{\sum_{j=1}^{q} |U_{g_j}|}, \tag{7}$$

$$CAP_j = \frac{\sum_{u \in U_{g_j}} Perf_u}{\sum_{j=1}^{q} \sum_{u \in U_{g_j}} Perf_u}, \tag{8}$$

Table 6. Group Proportion of All Users (*PAU*) and Contribution
to the Average Performance (*CAP*)

Dataset	Method	[0, 0.2]	(0.2, 0.4]	(0.4, 0.6]	(0.6, 0.8]	(0.8, 1]
				User Group		
TaFeng	*PAU*	68.8%	14.9%	8.5%	2.4%	5.5%
	G-TopFreq	53.4%	10.2%	10.4%	3.3%	22.7%
	P-TopFreq	12.3%	20.9%	21.9%	8.2%	36.8%
	GP-TopFreq	21.9%	18.8%	19.5%	7.3%	32.5%
	Dream	60.3%	9.7%	9.0%	2.2%	18.8%
	Beacon	59.0%	9.3%	9.4%	2.5%	19.9%
	CLEA	60.4%	9.3%	9.2%	2.3%	18.9%
	Sets2Sets	27.9%	18.2%	18.3%	6.4%	29.1%
	DNNTSP	37.4%	16.3%	16.0%	5.5%	24.7%
	TIFUKNN	24.2%	19.8%	19.8%	7.1%	29.0%
	UP-CF@r	18.4%	21.2%	21.4%	8.4%	30.8%
Dunnhumby	*PAU*	31.4%	19.6%	21.8%	14.3%	12.9%
	G-TopFreq	16.5%	18.6%	25.2%	18.1%	21.6%
	P-TopFreq	3.1%	14.8%	25.7%	22.6%	33.8%
	GP-TopFreq	4.3%	14.7%	25.5%	22.3%	33.3%
	Dream	16.8%	19.2%	24.9%	17.9%	21.2%
	Beacon	16.8%	18.7%	24.8%	17.9%	21.8%
	CLEA	15.4%	12.2%	20.0%	12.2%	40.1%
	Sets2Sets	4.2%	16.1%	25.1%	19.7%	34.9%
	DNNTSP	5.0%	15.7%	25.6%	22.5%	31.2%
	TIFUKNN	4.9%	15.1%	25.1%	22.0%	32.9%
	UP-CF@r	3.7%	15.3%	25.2%	22.3%	33.5%
Instacart	*PAU*	13.2%	14.9%	20.0%	21.9%	30.0%
	G-TopFreq	8.9%	12.3%	19.7%	24.0%	35.0%
	P-TopFreq	1.6%	7.8%	16.4%	23.7%	50.4%
	GP-TopFreq	1.8%	7.9%	16.4%	23.6%	50.3%
	Dream	9.2%	12.3%	19.8%	23.8%	34.8%
	Beacon	8.9%	12.4%	19.5%	23.5%	35.6%
	CLEA	7.0%	8.6%	14.1%	17.6%	52.6%
	Sets2Sets	2.0%	8.7%	15.8%	21.3%	52.2%
	DNNTSP	2.0%	8.1%	16.5%	23.3%	50.1%
	TIFUKNN	1.8%	8.0%	16.3%	23.6%	50.3%
	UP-CF@r	1.8%	8.0%	16.5%	23.5%	50.2%

where U_{g_j} denotes the set of users in group g_j, q denotes the number of user groups, and *Perf*$_u$ represents the method's performance w.r.t. user u. Note that the performance metric we analyze in this section is *Recall*@10, but similar phenomena can be observed for other metrics.

The results in terms of *PAU* and *CAP* are shown in Table 6. Under the ideal circumstances, the contribution to the average performance *CAP* of each user group should be equal to its proportion of all users *PAU*; this would allow us to use the average performance of a method as its overall performance and leave no user group behind. However, we can see that $CAP_{(0.8, 1]}$ is much

higher than $PAU_{(0.8,1]}$ and $CAP_{[0,0.2]}$ is much lower than $PAU_{[0,0.2]}$ for every NBR method (both repeat-biased methods and explore-biased methods) on all datasets. On the TaFeng dataset, only 5.5% of the users belong to group $(0.8, 1]$. However, their contribution to the average performance ranges from 18.8% to 36.8%. On the Dunnhumby dataset, 31.4% of the users belong to group $[0, 0.2]$, whereas the $CAP_{[0,0.2]}$ for repeat-biased methods (i.e., P-TopFreq, GP-TopFreq, Sets2Sets, DNNTSP, TIFUKNN, and UP-CF@r) only ranges from 3.1% to 5.0%. The results reflect that there might be a long-tail distribution w.r.t. the user's contribution to the average performance (i.e., few users contribute a large proportion to the performance), since the NBR task for different users might have different difficulty levels.

Given the previous observations, we construct a simple example to demonstrate the potential limitations of average performance. Assume we have two user groups (i.e., group g_a with 10 users and group g_b with only 1 user), where the NBR task for g_b is easier than g_a. Assume also that we have a baseline method M_b whose performance *Perf* can achieve 0.02 in group g_a and 0.4 in group g_b. We have another two optimized methods, M_α and M_β. Compared to baseline M_b, M_α can achieve 100% improvement in group g_a, and M_β can also achieve 100% improvement in group g_b at the cost of 50% reduction in group g_a. In this case, the cumulative improvement of M_α is $0.02 \times 10 = 0.2$, whereas the cumulative improvement of M_β is $0.4 \times 1 - 0.01 \times 10 = 0.3$. M_β is considered to be better than M_α since M_β's average performance is higher. However, we notice that M_α can improve the performance of 10 users, whereas M_β can only improve the performance of 1 user and at the detriment of the other 10 users. To sum up, the average performance has limitations to represent the performance of methods on the NBR task, and it might put users in a specific group at a disadvantage.[10] We should calculate the performance of each user group to have a comprehensive understanding of the NBR method.

5.7 Treatment Effect for Items with Different Frequencies

The NBR scenario can be thought of in terms of a two-sided market with items and users [8, 30, 38]. So far, we have analyzed the user-side performance from several aspects. In this section, we analyze treatment effects of NBR methods from the item side. Specifically, we investigate the relation between an item's exposure and its frequency in training labels (the ground-truth items of the training users) or test inputs (the historical items of the test users). As the item exposure in recommended baskets and the item frequency have different scales, we use the exposure and frequency of all items, respectively, to normalize them. To visualize the relation between an item's exposure and its frequency, we rank items according to their frequency and select the top-500 items. The frequency and the exposure distributions for different methods on the TaFeng dataset are shown in Figure 6.[11]

First, we observe the long-tail distribution w.r.t. the item exposure in all NBR methods; a small number of items get a large proportion of the total exposure. Surprisingly, a large proportion of items do not get any exposure in the baskets recommended by Dream, Beacon, and CLEA on the TaFeng dataset, which we consider to be sub-optimal from the item perspective. Second, the item exposure distributions of P-TopFreq, TIFUKNN, and UP-CF@r are more related to the frequency distribution of the test input than to the frequency distribution of the training labels. We believe that the repeat-biased nature of those algorithms, as well as the absence of training, results in

---

[10]In this article, to remain focused, we only analyze the repetition-exploration issue. However, there might be other factors (e.g., basket size, historical basket length) that also influence the difficulty level of the NBR problem.
[11]Experimental results on the Dunnhumby and Instacart datasets are provided in the appendix, and qualitatively similar patterns can be observed. We do not include G-TopFreq in this analysis, since it always recommends top-K items in the historical dataset.
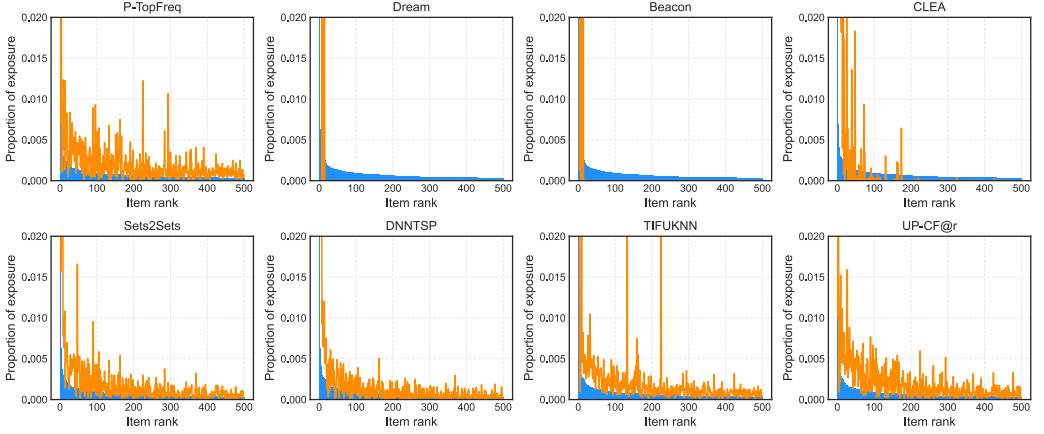
(a) Items ranked according to their frequency in the training labels
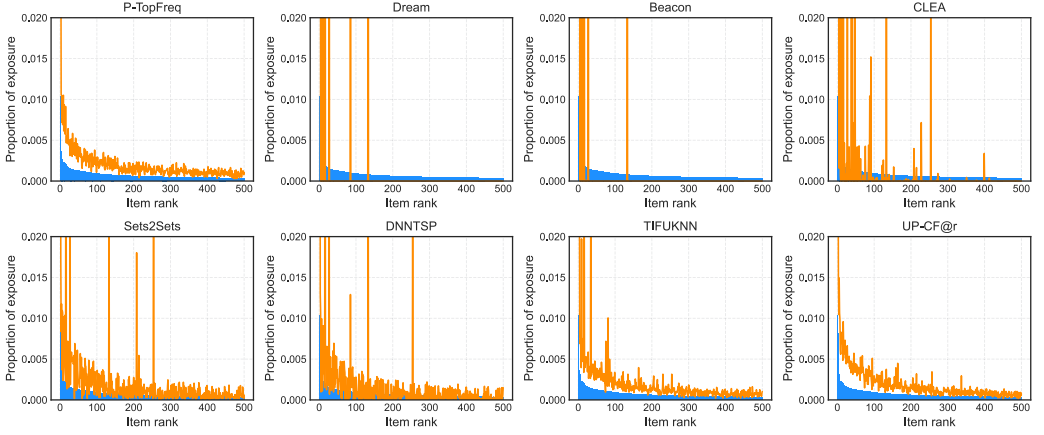
(b) Items ranked according to their frequency in the test inputs (i.e., historical baskets of test users)

Fig. 6. Treatment effect for items on the TaFeng dataset for eight NBR methods for items with different frequencies in training labels and testing inputs. The blue bar shows the frequency distribution, and the orange line denotes the exposure distribution.

recommendations that are strongly dependent on the items' frequency in historical baskets (i.e., on the test inputs). Third, in deep learning based methods (Dream, Beacon, CLEA, Sets2Sets, and DNNTSP), we can see that the distribution of items with high exposure shifts to the left, from Figure 6(b) to Figure 6(a). This result reflects the fact that an item's high exposure is more closely related to its high frequency in the training labels. To sum up, the item frequency distributions in the training labels and test inputs have a different impact on the item exposure of different NBR methods.

5.8 Upshot

Based on our second round of analyses of state-of-the-art NBR methods that we conducted with purpose-built metrics, we observe that there is a clear difficulty gap and tradeoff between the repetition task and the exploration task. As a rule of thumb, being biased toward the easier

repetition task is an important strategy that helps to boost the overall NBR performance. Deep learning based methods do not effectively exploit the repetition behavior. Indeed, they achieve a relatively good exploration performance, but they are not able to outperform the simple frequency baseline GP-TopFreq in several cases. Some recent state-of-the-art NBR methods are skewed toward the repetition task and outperform GP-TopFreq. However, the improvements they achieve are limited, especially considering the complexity and computational costs (e.g., for the training process [44] and for hyper-parameters search [16, 21]).

Moreover, current NBR methods usually focus on improving the overall performance, but they often fail to provide, or exploit, deeper insights into the components of their recommended baskets (skewed toward repetition or exploration).

Furthermore, different NBR methods have different treatment effects across different user groups, and the widely used average performance cannot fully evaluate the models' performance—for example, methods might achieve high overall performance at the detriment of a specific user group, which accounts for a large proportion of all users. From the item-side perspective, few items account for a large proportion of the total exposure in all NBR methods, and some NBR methods might only recommend a small set of items to users.

6 CONCLUSION

We have re-examined the reported performance gains of deep learning based methods for the NBR task over frequency-based and nearest neighbor based methods. We analyzed state-of-the-art NBR methods on the following seven aspects: (i) the overall performance on different scenarios, (ii) the basket components, (iii) the repeat and explore performance, (iv) the contribution of repetition and exploration to the overall performance, (v) the treatment effect for different user groups, (vi) the potential limitations of the average metrics, and (vii) the treatment effect for different items.

6.1 Main Findings

We arrived at several important findings. First, no state-of-the-art NBR method, deep learning based or otherwise, consistently shows the best performance across datasets. Compared to a simple frequency-based baseline, the improvements are modest or even absent. Second, there is a clear difficulty gap and tradeoff between the repeat task and the explore task. As a rule of thumb, being biased toward the easier repeat task is an important strategy that helps to boost the overall NBR performance. Third, some NBR methods might achieve better average overall performance at the detriment of a user group with a large proportion of users. Fourth, deep learning based methods do not effectively exploit repeat behavior. They indeed achieve relatively good explore performance but are not able to outperform the simple frequency-based baseline GP-TopFreq in terms of the relatively easy repetition task. Some state-of-the-art NBR methods are skewed toward the repeat task, and because of this they are able to outperform GP-TopFreq; however, their improvements are limited, especially considering their added complexity and computational costs.

6.2 Insights for NBR Model Evaluation

Our work highlights the following important guidelines that practitioners and researchers working on NBR should follow when evaluating or designing an NBR model. First, use a diverse set of datasets for evaluation, with different ratios of repeat items and explore items. Second, use GP-TopFreq as a baseline when evaluating NBR methods. Third, apart from the conventional accuracy-based metrics, consider the newly introduced repeat and explore metrics, $Recall_{rep}$, PHR_{rep}, $Recall_{expl}$, and PHR_{expl}, as a set of fundamental metrics to understand the performance of NBR methods. Fourth, the $RepR$ and $ExplR$ statistics should be included to understand what kind

of items shape the recommended baskets. Fifth, calculate the performance of each user group to get a comprehensive understanding of the NBR methods.

6.3 Insights for NBR Model Design

From the analysis of this article, apart from the difficulty imbalance between the repetition and exploration task, we should also be aware that the repetition recommendation task and exploration recommendation task have different characteristics. For instance, the repetition recommendation task focuses on predicting whether historical items will be repurchased or not, where the frequency and recency of historical items are quite important, and the exploration recommendation task focuses on inferring *explore items* from a much bigger search space via modeling item-to-item correlations, which deep learning methods might be good at. Therefore, just blindly designing complex NBR models without considering the difference between repetition and exploration might be sub-optimal.

We believe that it is better to separate the repetition recommendation and exploration recommendation in the NBR task (e.g., using frequency and recency to address the repetition task, and using NN-based models to model item-to-item correlations), which not only allows us to address repetition and exploration according to their respective characteristics but also offers the flexibility of controlling repetition and exploration in the recommended basket. Additionally, believe that future NBR methods should be able to combine repetition and exploration based on users' preferences.

6.4 Limitations

One of the limitations of this study is that we did not consider the training and inference execution time in the article, which is important for the real-world value of methods used for NBR [4]. We use the original implementations of NBR methods to check their reproducibility and avoid potential mistakes that may come with re-implementations; however, the original implementations are based on different frameworks, which leads to an inability to make a fair execution time comparison. A second limitation is that we only follow the widely used binary definition of repeat items and explore items but do not consider a more fine-grained formalization based on the frequency of historical items, which would allow for a more flexible definition of repetition and analysis. A further limitation is that we only considered the short-term utility of NBR methods: will users be satisfied with their next basket? Limited by our experimental setup, where we replay users' past behavior, we have ignored any potential long-term effects of having a strong focus on short-term utility by emphasizing repeat items as opposed to, for instance, long-term engagement that, likely, benefits from a certain degree of exploration so as to enable surprise and discovery.

6.5 Future Work

Obvious avenues for future work include addressing the limitations summarized previously. Another important line of future work concerns the use of domain-specific knowledge, either concerning complementarity or substitutability of items or concerning hierarchical relations between items, both of which would allow one to consider more semantically informed notions of repeat consumption behavior [3] for NBR purposes. In addition, our focus in this article has been on users (i.e., in the sense that we compared methods that produce a basket for a given user), and it would be interesting to consider repetition and exploration aspects of the reverse scenario [27] (i.e., given an item, who are the users to whose baskets this item can best be added?). Finally, even though we focused on NBR, it would be interesting to contrast our outcomes with an analysis of repeat and explore behavior in traditional sequential recommendation scenarios.

SUPPLEMENTARY MATERIALS

To facilitate reproducibility of our work, we share the following materials at https://github.com/liming-7/A-Next-Basket-Recommendation-Reality-Check): (i) source code and datasets, (ii) descriptions of different dataset formats, and (iii) pipelines for running our experiments and obtaining our results.

A APPENDIX

A.1 Additional Plots

We include additional plots to supplement Figure 3 (Figure 7) and Figure 6 (Figures 8 and 9), respectively.

A.2 Reproducibility

To facilitate reproducibility of the results in this article, our online repository, which can be found at https://github.com/liming-7/A-Next-Basket-Recommendation-Reality-Check, contains the following resources: (i) source code and datasets, (ii) descriptions of different dataset format, and (iii) pipelines about how to run and get results.
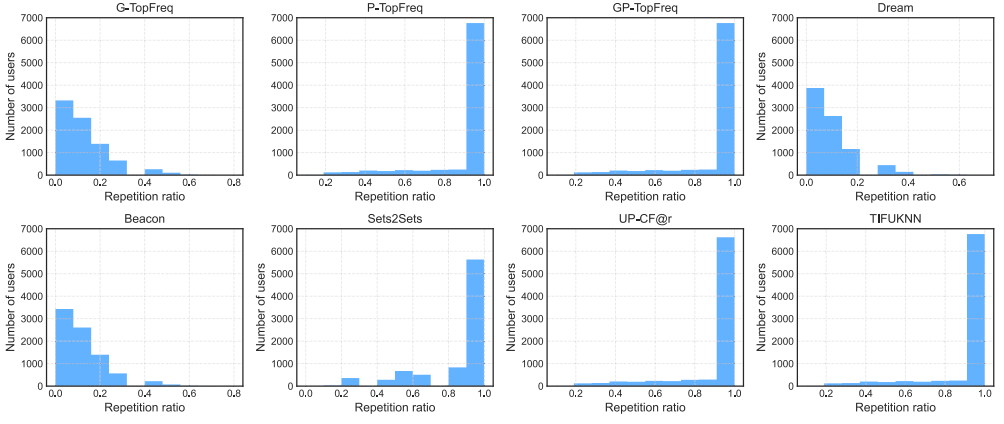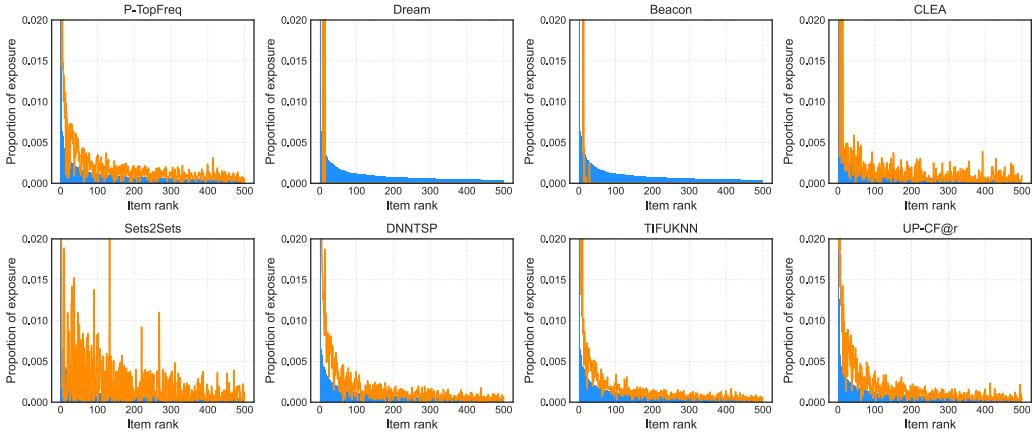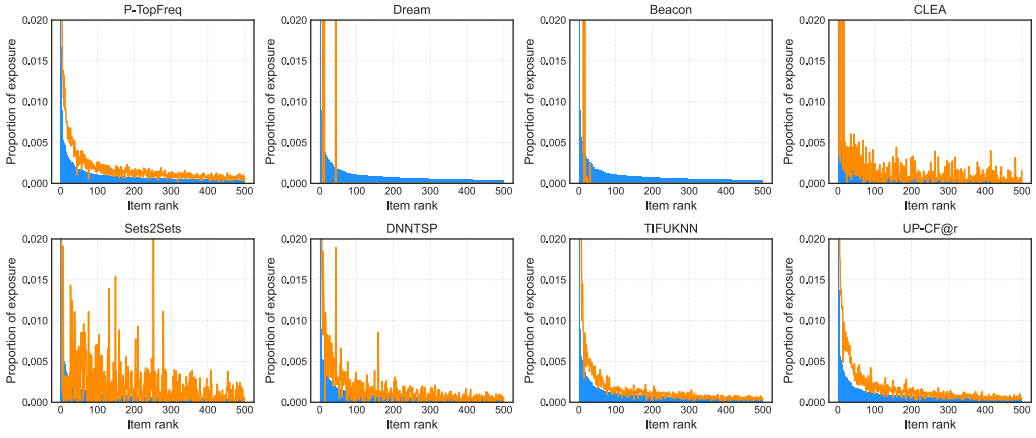
(a) TaFeng
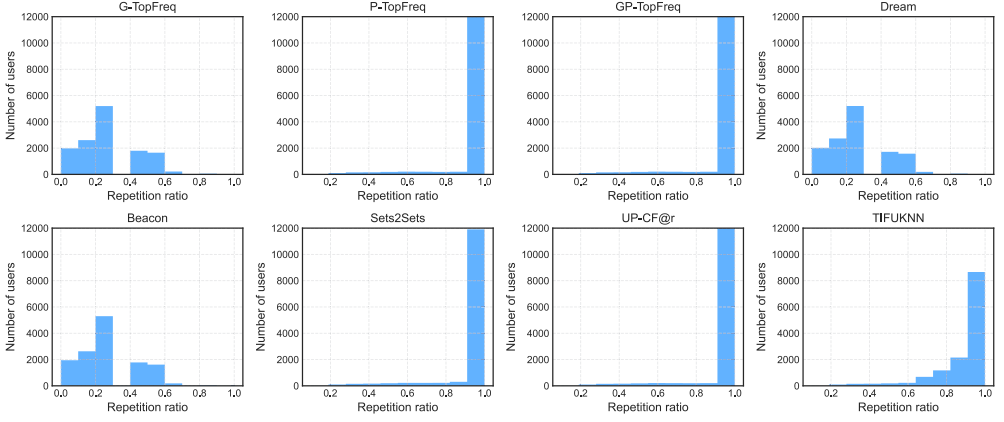
(b) Dunnhumby
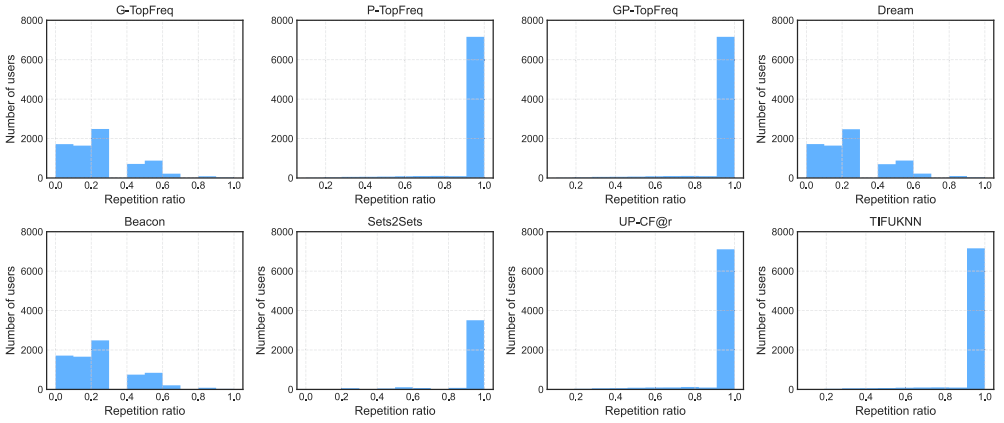
(c) Instacart

Fig. 7. Distribution of the repetition ratio *RepR* of recommended baskets on the TaFeng dataset for eight NBR methods (G-TopFreq, P-TopFreq, GP-TopFreq, Dream, Beacon, Sets2Sets, UP-CF@r, TIFUKNN).
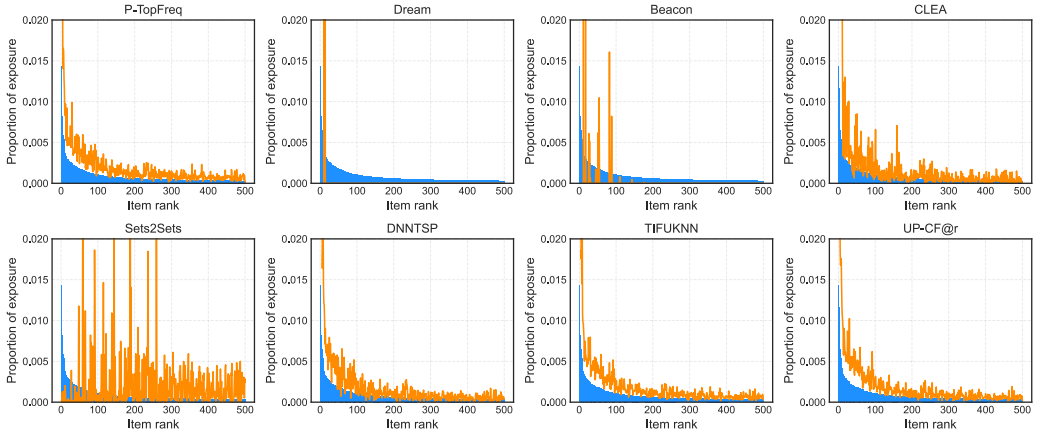
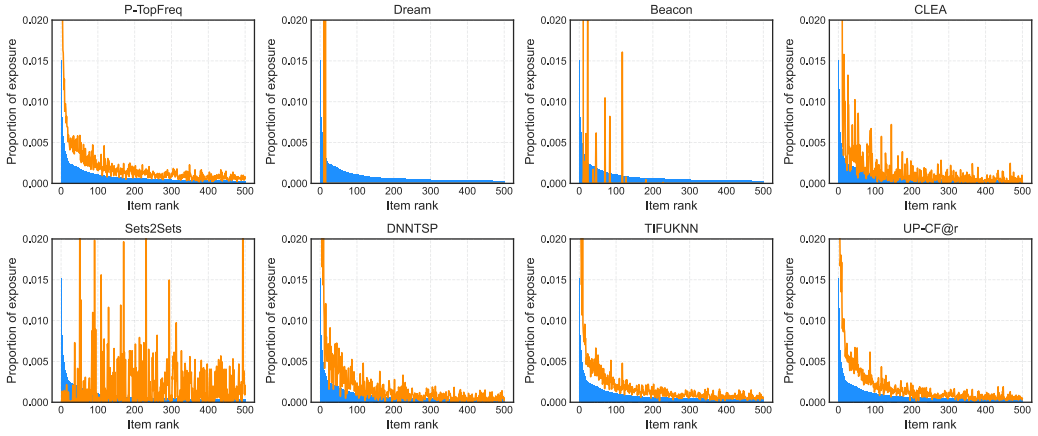(a) Items ranked according to their frequencies in the training labels

(b) Items ranked according to their frequencies in the testing inputs (i.e., historical baskets of testing users)

Fig. 8. Treatment effect for items on the Dunnhumby dataset for eight NBR methods for items with different frequencies in training labels and testing inputs. The blue bar shows the frequency distribution, and the orange line denotes the exposure distribution.

(a) Items ranked according to their frequencies in the training labels

(b) Items ranked according to their frequencies in the testing inputs (i.e., historical baskets of testing users)

Fig. 9. Treatment effect for items on the Instacart dataset for eight NBR methods for items with different frequencies in training labels and testing inputs. The blue bar shows the frequency distribution, and the orange line denotes the exposure distribution.

ACKNOWLEDGMENTS

REFERENCES

[1] Mehmet Akcay, Ismail Sengor Altingovde, Craig Macdonald, and Iadh Ounis. 2017. On the additivity and weak baselines for search result diversification research. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, New York, NY, 109–116.

[2] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. 2014. The dynamics of repeat consumption. In *Proceedings of the 23rd International Conference on World Wide Web*. 419–430.

[3] 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 (SIGIR'22)*. ACM, New York, NY, 1240–1250.

[4] Mozhdeh Ariannezhad, Ming Li, Sebastian Schelter, and Maarten de Rijke. 2023. A personalized neighborhood-based model for within-basket recommendation in grocery shopping. In *Proceedings of the 16th International Conference on Web Search and Data Mining (WSDM'23)*. ACM, New York, NY.

[5] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. 2009. Improvements that don't add up: Ad-hoc retrieval results since 1998. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. ACM, New York, NY, 601–610.

[6] 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 *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'18)*. 1201–1204.

[7] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. 2019. A review on deep learning for recommender systems: Challenges and remedies. *Artificial Intelligence Review* 52 (2019), 1–37.

[8] Arpita Biswas, Gourab K. Patro, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2021. Toward fair recommendation in two-sided platforms. *ACM Transactions on the Web* 16, 2 (2021), 1–34.

[9] Timo Breuer, Nicola Ferro, Norbert Fuhr, Maria Maistro, Tetsuya Sakai, Philipp Schaer, and Ian Soboroff. 2020. How to measure the reproducibility of system-oriented IR experiments. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20)*. ACM, New York, NY, 349–358.

[10] Jun Chen, Chaokun Wang, and Jianmin Wang. 2015. Will you "reconsume" the near past? Fast prediction on short-term reconsumption behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[11] Jun Chen, Chaokun Wang, Jianmin Wang, and S. Yu Philip. 2016. Recommendation for repeat consumption from user implicit feedback. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (2016), 3083–3097.

[12] Yongjun Chen, Jia Li, Chenghao Liu, Chenxi Li, Markus Anderle, Julian McAuley, and Caiming Xiong. 2021. Modeling dynamic attributes for next basket recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*.

[13] Matt Crane. 2018. Questionable answers in question answering research: Reproducibility and variability of published results. *Transactions of the Association for Computational Linguistics* 6 (2018), 241–252.

[14] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems* 39, 2 (2021), 1–49.

[15] 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.

[16] 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.

[17] Nicola Ferro and Diane Kelly. 2018. SIGIR initiative to implement ACM artifact review and badging. *SIGIR Forum* 52, 1 (June 2018), 4–10.

[18] Ming C. Hao, Meichun Hsu, Umeshwar Dayal, Shu Feng Wei, Thomas Sprenger, and Thomas Holenstein. 2001. Market basket analysis visualization on a spherical surface. In *Visual Data Exploration and Analysis VIII*, Vol. 4302. International Society for Optics and Photonics, SPIE, 227–233.

[19] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. 2008. A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. ACM, New York, NY, 1419–1420.

[20] 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.

[21] 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'20)*. 1071–1080.

[22] Dietmar Jannach, Gabriel de Souza P. Moreira, and Even Oldridge. 2020. Why are deep learning models not consistently winning recommender systems competitions yet? A position paper. In *Proceedings of the Recommender Systems Challenge 2020 (RecSysChallenge'20)*. ACM, New York, NY, 44–49.

[23] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining*. 197–206.

[24] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1748–1757.

[25] 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.

[26] Youfang Leng, Li Yu, Jie Xiong, and Guanyu Xu. 2020. Recurrent convolution basket map for diversity next-basket recommendation. In *Proceedings of the International Conference on Database Systems for Advanced Applications*. 638–653.

[27] Ming Li, Mozhdeh Ariannezhad, Andrew Yates, and Maarten de Rijke. 2023. Who will purchase this item next? Reverse next period recommendation in grocery shopping. *ACM Transactions on Recommender Systems*. Accepted subject to major revisions.

[28] Jimmy Lin. 2019. The neural hype and comparisons against weak baselines. *SIGIR Forum* 52, 2 (Jan. 2019), 40–51.

[29] Harrie Oosterhuis and Maarten de Rijke. 2019. Optimizing ranking models in an online setting. In *Proceedings of the 41st European Conference on Information Retrieval (ECIR'19)*. 382–396.

[30] Gourab K. Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2020. FairRec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of the 2020 Web Conference*. 1194–1204.

[31] 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)*. 859–868.

[32] Troy Raeder and Nitesh V. Chawla. 2011. Market basket analysis with networks. *Social Network Analysis and Mining* 1, 2 (2011), 97–113.

[33] 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.

[34] 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'20)*. 1439–1448.

[35] Niek Tax, Sander Bockting, and Djoerd Hiemstra. 2015. A cross-benchmark comparison of 87 learning to rank methods. *Information Processing & Management* 51, 6 (2015), 757–772.

[36] Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 5th ACM Conference on Recommender Systems*. 109–116.

[37] Ellen M. Voorhees and Donna K. Harman (Eds.). 2005. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, MA.

[38] Lequn Wang and Thorsten Joachims. 2021. User fairness, item fairness, and diversity for rankings in two-sided markets. In *Proceedings of the 44th ACM SIGIR International Conference on Theory of Information Retrieval*. 23–41.

[39] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for next basket recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. 403–412.

[40] 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.

[41] 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.

[42] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. 2019. Critically examining the "neural hype": Weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. ACM, New York, NY, 1129–1132.

[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 (SIGIR'16)*. 729–732.

[44] 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.