CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation

Shengyu Zhang*
Zhejiang University, China
sy_zhang@zju.edu.cn

Dong Yao*
Zhejiang University, China
yaodongai@zju.edu.cn

Zhou Zhao†
Zhejiang University, China
zhaozhou@zju.edu.cn

Tat-seng Chua
National University of Singapore, Singapore
dcscts@nus.edu.sg

Fei Wu†
Zhejiang University, China
wufei@zju.edu.cn

ABSTRACT

Learning user representations based on historical behaviors lies at the core of modern recommender systems. Recent advances in sequential recommenders have convincingly demonstrated high capability in extracting effective user representations from the given behavior sequences. Despite significant progress, we argue that solely modeling the observational behaviors sequences may end up with a brittle and unstable system due to the noisy and sparse nature of user interactions logged. In this paper, we propose to learn accurate and robust user representations, which are required to be less sensitive to (attack on) noisy behaviors and trust more on the indispensable ones, by modeling counterfactual data distribution. Specifically, given an observed behavior sequence, the proposed CauseRec framework identifies dispensable and indispensable concepts at both the fine-grained item level and the abstract interest level. CauseRec conditionally samples user concept sequences from the counterfactual data distributions by replacing dispensable and indispensable concepts within the original concept sequence. With user representations obtained from the synthesized user sequences, CauseRec performs contrastive user representation learning by contrasting the counterfactual with the observational. We conduct extensive experiments on real-world public recommendation benchmarks and justify the effectiveness of CauseRec with multi-aspects model analysis. The results demonstrate that the proposed CauseRec outperforms state-of-the-art sequential recommenders by learning accurate and robust user representations .

CCS CONCEPTS

• **Information systems → Recommender systems**.

*These authors contributed equally to this work.
†Corresponding Authors.

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

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462908

KEYWORDS

Sequential Recommendation, User Modeling, Contrastive Learning, Counterfactual Representation

ACM Reference Format:
Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-seng Chua, Fei Wu. 2021. CauseRec: Counterfactual User Sequence Synthesis for Sequential 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, 11 pages. https://doi.org/10.1145/3404835.3462908

1 INTRODUCTION

Due to the overwhelming data that people are facing on the Internet, personalized recommendation has become vital for retrieving information and discovering content. Accurately characterizing and representing users plays a vital role in a successful recommendation framework. Since users' historical interactions are sequentially dependent and by nature time-evolving, recent advances [40, 42, 43, 48, 57–60, 63, 71, 74, 82] pay attention to sequential recommendation, which captures the current and recent preference by exploiting the sequentially modeled user-item interactions.

A sequential recommender aims to predict the next item a user might interact with based on the historical interactions. The challenging and open-ended nature of sequence modeling lends itself to a variety of diverse models. Traditional methods mainly exploit Markov chains [15] and factorization machines [22, 49] to capture lower-order sequential dependencies. Following these works, the higher-order Markov Chain and RNN (Recurrent Neural Network) [18, 21, 66] are proposed to model the complex high-order sequential dependencies. More recently, MIND is proposed to transform the historical interactions into multiple interest vectors using the capsule network [51]. ComiRec [5] differs from MIND by leveraging the attention mechanism and introducing a factor to control the balance of recommendation accuracy and diversity.

Despite significant progress made with these frameworks, there are some challenges demanding further explorations. A vital challenge comes from the noisy nature of implicit feedback. Due to the ubiquitous distractions that may affect the users' first impressions (such as caption bias [39], position bias [24], and sales promotions), there are inconsistencies between users' interest and their clicking behaviors, known as the natural noise [45]. Another challenge

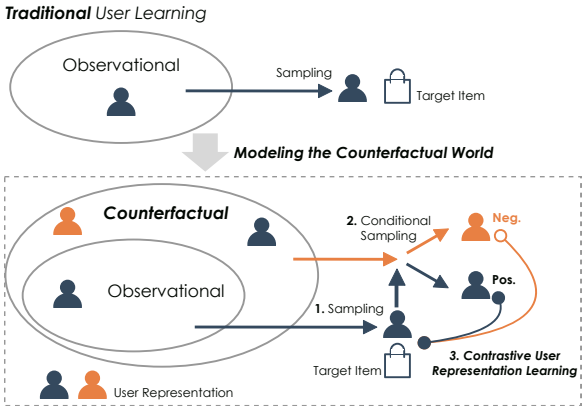


Figure 1: An illustration of the proposed contrastive user representation learning by modeling the counterfactual world (below), compared with most traditional approaches that solely model the observational user sequences (above).

relates to the deficiency of existing methods in confronting data sparsity problem in recommender systems where users in general only interact with a limited number of items compared with the item gallery which can easily reach 100 million in large live systems. Therefore, solely modeling the observational behavior sequences that can be both sparse and noisy may end up with a brittle system that is less satisfactory. To this end, learning **accurate** and **robust** users' user representations is essential for recommender systems.

In this paper, we propose ***C***ounterf***a***ctual ***U***ser ***Se***quence Synthesis for Sequential Recommendation, abbreviated as **CauseRec**. The essence of CauseRec in confronting the data sparsity problem is to model the counterfactual data distribution rather than the observational sparse data distribution where the latter can be a subset of the former one, as shown in Figure 1. We mainly aim to answer the counterfactual question, "what the user representation would be if we intervene on the observed behavior sequence?". Specifically, given the observed behavior sequence, we identify indispensable/dispensable concepts at both the fine-grained item level and the abstract interest level. A concept indicates a certain aspect of user interest/preference. We perform counterfactual transformations on both the item-level and the interest-level user concept sequences. We obtain counterfactually positive user representation by modifying dispensable concepts, and counterfactually negative user representation by replacing indispensable concepts. To learn **accurate** and **robust** user representations, we propose to conduct contrastive learning between: 1) the observational and the counterfactual user representations; and 2) the user representations and the target items. Contrast with such out-of-distribution hard negatives potentially makes the learned representations ***robust*** since they are less sensitive to dispensable/noisy concepts. Contrast with such out-of-distribution positives potentially makes the learned representations ***accurate*** since they will trust more on the indispensable concepts that are better representing user's interest.

We conduct in-depth experiments to validate the effectiveness of the proposed CauseRec architectures on various public recommendation datasets. With a naive deep candidate generation (or matching) architecture as the baseline method, CauseRec outperforms SOTA sequential recommenders for deep candidate generation. We conduct comprehensive model analysis to uncover how different building blocks and hyper-parameters affect the performance of CauseRec. Case studies further demonstrate that CauseRec can help learn accurate user representations. To summarize, this paper makes the following key contributions:

- We propose to model the counterfactual data distribution (besides the observational data distribution) to confront the data sparsity problem for recommendation.
- We devise the CauseRec framework which learns *accurate* and *robust* user representations with counterfactual transformations on both fine-grained item-level and abstract interest-level, and with various contrastive objectives.
- We conduct extensive experiments and show that with a naive deep candidate generation architecture as the baseline, CauseRec can outperform SOTA sequential recommenders.

2 RELATED WORKS

2.1 Sequential Recommendation

Sequential recommendation can be traced back to leveraging Markov-chain [14, 15] and factorization machines [22, 49]. To capture long-term and multi-level cascading dependencies, deep learning based techniques (*e.g.*, RNNs [10, 21, 47, 66] and CNNs [55, 75]) are incorporated into sequential modeling. DNNs are known to have enticing representation capability and have the natural strength to capture comprehensive relations [76] over different entities (*e.g.*, items, users, interactions). Recently, there are works that explore advanced techniques, *e.g.*, memory networks [53], attention mechanisms [56, 79], and graph neural networks [9, 26, 31, 36, 81] for sequential recommendation [6, 23, 29, 54, 61, 67, 72]. Typically, MIND [32] adopts the dynamic routing mechanism to aggregate users' behaviors into multiple interest vectors. ComiRec [5] differs from MIND by leveraging the attention mechanism for user representations and proposes a factor for the trade-off between recommendation diversity and accuracy. Different from the above works that solely model the observational user sequences, we take a step further to model the counterfactual data distributions. By contrasting the user representations of the observation with the counterfactual, we aim to learn user encoders that can better confront out-of-distribution user sequences and learn accurate and robust user representations.

2.2 Contrastive Learning for Recommendation

A growing number of attempts have been made to exploit the complementary power of self-supervised learning (*e.g.*, contrastive learning) and deep learning, with domains varying from computer vision [12, 17, 68], natural language generation [7, 77], to graph embedding [46]. However, how to consolidate the merits of contrastive learning into recommendation remains largely unexplored in the literature. Recently, Sun *et al.*[33] adopt noise contrastive estimation [16] to transfer the knowledge from a large natural language corpus to recommendation-specific content that is sparse on long-tail publishers and thus learning effective word representations. CLRec [83] bridges the theoretical gap between contrastive learning objective and traditional recommendation objective, *e.g.*, sampled softmax loss, as well as more advanced inverse propensity weighted

(IPW) loss. They show that directly performing contrastive learning can help to reduce exposure bias. CP4Rec [69] and S^3-Rec [84] integrates Bert structure and contrastive learning objective for user pretraining, which require a fine-tuning stage. Compared with these works, we design model-agnostic and non-intrusive frameworks that help any baseline model learn more effective user representations in an end-to-end manner. Such representations are more accurate and robust by contrasting the original user representation with counterfactually positive samples and counterfactually negatives samples.

2.3 Counterfactual for Recommendation

Causality and counterfactual reasoning have attracted great attentions in various domains [11, 78, 80]. Previous counterfactual frameworks in recommendation focus on debiasing the learning-to-rank problems. A rigorous counterfactual learning framework, *i.e.*, PropDCG [1], is proposed to overcome the distorting effect of presentation bias. The position bias and the clickbait issue are investigated in [2, 64] and [62], respectively. The Inverse Propensity Score [52, 70] method obtains unbiased estimation by sample re-weighting based on the likelihood of being logged. Another line of works encapsulates the uniform data into recommendation by learning imputation models [73], computing propensity [52], using knowledge distillation [13, 37], and directly modeling the uniform data [4, 28, 38, 50]. Different from these works, we focus on denoising user representation learning and considers the retrospect question, *i.e.*, 'what the user representation would be if we intervene on the observed behavior sequence?'. Technically, we propose several counterfactual transformations based on the identification of indispensable/dispensable concepts and devise several contrasting objectives for learning accurate and robust user representations.

3 METHODS

3.1 Problem Formulation

In the view of sequential recommendation, datasets can be formulated as $\mathcal{D} = \{(x_{u,t}, y_{u,t})\}_{u=1,2,\ldots,N, t=1,2,\ldots,T_u}$, where $x_{u,t} = \{y_{u,1:(t-1)}\}$ denotes a user's historical behaviors prior to the tth behavior $y_{u,t}$ and arranged in a chronological order, and T_u denotes the number of behaviors for the user u. The goal of sequential recommendation is to predict the next item $y_{u,t}$ given the historical behaviors $x_{u,t}$, which can be formulated as modeling the probability of all possible items:

$$p\left(y_{u,t} = y | x_{u,t}\right), \tag{1}$$

We will drop the sub-scripts occasionally and write (x, y) in place of $(x_{u,t}, y_{u,t})$ for simplicity. Let $\mathcal{X}$ denote the set of all possible click sequences, i.e. $x \in \mathcal{X}$ and each $y \in \mathcal{Y}$ represent a clicked item, while $\mathcal{Y}$ is the set of all possible items.

Since the number of items $|\mathcal{Y}|$ can easily reach 100 million, industrial recommender systems consist typically of two phases, *i.e.*, the matching phase and the ranking phase, due to concerns on system latency. The matching (also called deep candidate generation) phase focuses on retrieving Top N candidates for each user, while the ranking phase further sorts the N candidates by typically considering more fine-grained user/item features and incorporating complex modeling architectures. In this paper, we mainly conduct experiments in the matching stage (*e.g.*, comparing with SOTA matching models).

3.2 A Naive Matching Baseline

The paradigm of a matching model includes a user encoder $f_\theta(x) \in \mathbb{R}^d$, which takes the user's historical behavior sequence as input and output one (or more) dense vector representing the user's interests, and an item encoder $g_\theta(y) \in \mathbb{R}^d$, that represents the items in the same vector space as the user encoder. We denote all the trainable parameters in the system as θ, which includes the parameters in f_θ and g_θ. With the learned encoders and the extracted item vectors, *i.e.*, $\{g_\theta(y)\}_{y \in \mathcal{Y}}$, a k-nearest-neighbor search service, e.g., Faiss [27], will be deployed for Top-N recommendation. Specifically, at serving time, an arbitrary user behavior sequence x will be transformed into a vectorial representation $f_\theta(x)$ and top N items with the largest matching scores will be retrieved as Top-N candidates. Such matching scores are typically computed as inner product $\phi_\theta(x, y) = \langle f_\theta(x), g_\theta(y) \rangle$ or cosine similarity. In a nutshell, the learning procedure can be formulated as the following maximum likelihood estimation:

$$\arg\min_\theta \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} -\log p_\theta(y \mid x), \tag{2}$$

$$\text{where } p_\theta(y \mid x) = \frac{\exp \phi_\theta(x, y)}{\sum_{y' \in \mathcal{Y}} \exp \phi_\theta(x, y')}. \tag{3}$$

In the matching phase, it can be infeasible to sum over all possible items y' as in the denominator. Here we adopt the sample softmax objective [3, 25]. To demonstrate the effectiveness of the proposed CauseRec architecture, we utilize a naive framework as the baseline. Specifically, the *item* encoder $g_\theta(y)$ is a plain lookup embedding matrix where the nth vector represents the item embedding with item id n. The *user* encoder $f_\theta(x)$ aggregates the embeddings of historically interacted items using global average pooling and then transforms the aggregated embedding into the same embedding space as item embeddings using multi-layer perceptrons (MLP):

$$f_\theta(x) = \text{MLP}(\frac{1}{t-1} \sum_{i=1}^{t-1} g_\theta(y_i)). \tag{4}$$

3.3 The CauseRec Architecture

In this section, we give an brief illustration on the intuition and overall schema/pipeline of the CauseRec architecture, which is depicted in Figure 2, and introduce the building blocks in detail.

3.3.1 Overall Schema. The essence of CauseRec is to answer the retrospect question, 'what the user representation would be if we intervene on the observed behavior sequence?' The counterfactual transformation in CauseRec relates to the 'intervention on the observed behavior sequence.' For answering 'what the user representation would be,' we introduce an important inductive bias that makes the intervention work as expected. Specifically, we first identify indispensable/dispensable concepts in the historical behavior sequence. An indispensable concept indicates a subset of one behavior sequence that can jointly represent a meaningful aspect of the user's interest. A dispensable concept indicates a noisy subset that is less meaningful/important in representing an aspect of interest. We introduce the details in Section 3.3.2.

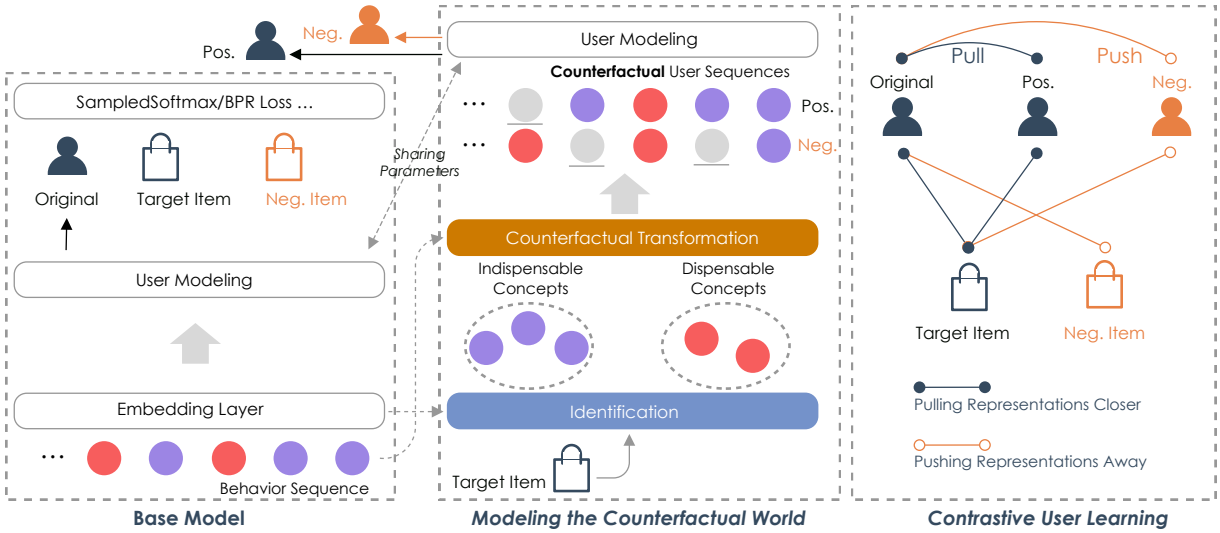


Figure 2: Schematic of the proposed CauseRec-Item framework.

Given the identified concepts, a representative counterfactual transformation is designed to build out-of-distribution counterfactual user sequences. Here comes the inductive bias, *i.e.*, counterfactual sequences constructed by replacing the dispensable concepts in the original user sequence should still have similar semantics to the original one. Here *semantics* refer to the characteristics of user interests/preferences. Therefore, replacing indispensable concepts in the original user sequence should incur a preference deviation from the resulted user representation to the original user representation. We denote these resulted user representations as counterfactually *negative* user representations. We note that such negatives are hard negatives where *hard* refers to that other dispensable concepts stay the same as the original user sequence and *negatives* means that the semantics of the user sequence should be different. In contrast, replacing dispensable concepts in the original user sequence should incur no preference change in representations. We denote these resulted user representations as counterfactually *positive* user representations. Different contrastive learning objectives are proposed to learn accurate and robust user representations that are less sensitive to the (attack on the) dispensable/noisy concepts and that trust more on the indispensable concepts that better represent the user's interest. Details can be found in Section 3.3.5.

3.3.2 Identification of Indispensable/Dispensable concepts. To identify indispensable/dispensable concepts, we propose to first extract concept proposals and compute the proposal scores.

Item-level Concepts. Inspired by instance discrimination [17], a straightforward while workable solution is to treat each item in the behavior sequence as an individual concept since each item has its unique fine-grained characteristics. In this way, we obtain the concept sequence $\mathbf{C} = \mathbf{X} \in \mathbb{R}^{t \times d}$, where $\mathbf{X} = g_\theta(x_{u,t+1})$ denotes the vectorial representations of the behavior sequence. In essence, concept scores indicate to what extent these concepts are important to represent the user's interest. Since there is no groundtruth for one user's real interest, we use the target item y_{t+1} as the indicator:

$$p_i^{item} = \phi_\theta(\mathbf{c}_i, \mathbf{y}), \tag{5}$$

where $\mathbf{c}_i$ indicates the representation of ith concept in $\mathbf{C}$, and $\mathbf{y}$ indicates the representation of the target item. ϕ_θ is the similarity function, and we empirically use dot product for its effectiveness in the experiment. p_i^{item} is thus the score for the ith concept.

Interest-level Concepts. However, such a solution may incur redundancy in concepts since some items may share similar semantics, and might deteriorate the capability of modeling higher-order relationships between items. To this end, besides the item-level concepts, we introduce interest-level concepts by leveraging the attention mechanism [56] to extract interest-level concepts. Formally, $\mathbf{X} \in \mathbb{R}^{t \times d}$, we obtain the following attention matrix:

$$\mathbf{A} = \text{softmax}\left(\mathbf{W}_2 \tanh\left(\mathbf{W}_1 \mathbf{X}^\top\right)\right)^\top, \tag{6}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_a \times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{K \times d_a}$ are trainable transformation matrices. K is thus the number of concepts that is pre-defined. $\mathbf{A}$ is of shape $\mathbb{R}^{t \times K}$. We obtain the concept sequence as the following:

$$\mathbf{C} = \mathbf{A}^\top \mathbf{X}, \tag{7}$$

Since interest-level concepts and the target item are not naturally embedded in the same space, we compute the concept score by the weighted sum of item-level scores:

$$p^{interest} = \mathbf{A}^\top \phi_\theta(\mathbf{X}, \mathbf{y}). \tag{8}$$

For both item-level concepts and interest-level concepts, we treat the top half concepts with the highest scores as indispensable concepts and the remaining half concepts as dispensable concepts. This strategy is mainly designed to prevent the number of indispensable or dispensable concepts from being too small. We leave finding more effective solutions as future works as illustrated in Section 5.

3.3.3 Counterfactual Transformation. The proposed counterfactual transformation aims to construct out-of-distribution user sequences by transforming the original user sequence for one user. We here

use ***user sequence*** to generally denote the concept sequence, which can be either the commonly known item-level concept sequence, *i.e.*, the original user behavior sequence, or the interest-level concept sequence. Based on the inductive bias described in Section 3.3.1, we propose to replace the identified indispensable/dispensable concepts at the rate of r_{rep} to construct counterfactually negative/positive user sequences, respectively. We note that directly dropping indispensable/dispensable concepts also seems feasible, but replacement has the advantage of not affecting overall sequence length and the relative positions of remaining concepts. We maintain a first-in-first-out queue as a concept memory for each level and use dequeued concepts as substitutes. We enqueue the concepts extracted from the current mini-batch. We denote the user sequence with indispensable/dispensable concepts being replaced as counterfactually negative/positive user sequence.

3.3.4 User Encoders. We note that item-level concept representations can be trained along with the item embedding matrix in most recommendation frameworks. Therefore, the above item-level concept identification and counterfactual transformation processes can be performed without any modification on the user encoder in the original baseline model, *i.e.*, a model-agnostic and non-intrusive design. We denote the architecture solely considering item-level concepts as CauseRec-Item. CauseRec-Item obtains counterfactually positive/negative user representations $\{\mathbf{x}^{+,m}\}_{m=1,\ldots,M}/\{\mathbf{x}^{-,n}\}_{n=1,\ldots,N}$ from counterfactual item-level concept sequences using the original user encoder f_θ.

We denote the architecture solely considering interest-level concepts as CauseRec-Interest. Different from CauseRec-Item, interest-level concepts are constructed with learnable parameters, *i.e.*, $\mathbf{W}_1$ and $\mathbf{W}_2$ in Equation 6. Therefore, CauseRec-Interest is an intrusive design, and the inputs to the user encoder should be the interest-level concept sequence rather than the behavior sequence at the item-level. We note that there are no further modifications, and the architecture of the user encoder can stay the same as in the original baseline model. CauseRec-Interest obtains counterfactually positive/negative user representations from counterfactual interest-level concept sequence using the original user encoder f_θ.

We denote the architecture that considers counterfactual transformation on both the item-level concept sequence and the interest-level concept sequence as CauseRec-H(ierarchical). CauseRec-H is also an intrusive design with interest-level concepts as the inputs of the user encoder. Different from CauseRec-Interest, CauseRec-H further considers counterfactual transformations performed on item-level concepts. The counterfactually transformed item-level sequence will be forwarded to construct interest-level concept sequence using Equation 6-7. We note that counterfactual transformations will not be performed on these two levels simultaneously, which might introduce unnecessary noises. In other words, each counterfactual user representation is constructed with transformation on sequence solely from one level.

3.3.5 Learning Objectives. Besides the original recommendation loss $\mathcal{L}_{matching}$ described near Equation 2, we propose several contrastive learning objectives that are especially designed for learning accurate and robust user representations.

Contrast between Counterfactual and Observation. As discussed in Section 3.3.1, a ***robust*** user representation should be less sensitive to (possible attack on) dispensable concepts. Therefore, the user representations learned from counterfactual sequences with indispensable concepts transformed should be intuitively pushed away from the original user representation. Similar in spirit, an ***accurate*** representation should trust more on indispensable concepts. Therefore, user representations learned from counterfactual sequences with dispensable concepts transformed should be intuitively pulled closer to the original user representation. Under these intuitions, we derive inspiration from the recent success of contrastive learning in CV [17, 68] and NLP [7], we use triplet margin loss to measure the relative similarity between samples:

$$\mathcal{L}_{co} = \sum_{m=1}^{M}\sum_{n=1}^{N} \max\left\{d\left(\mathbf{x}^{q}, \mathbf{x}^{+,m}\right) - d\left(\mathbf{x}^{q}, \mathbf{x}^{-,n}\right) + \Delta_{\text{co}}, 0\right\}, \quad (9)$$

where $\mathbf{x}^q$ denotes the original user representation. We set the distance function d as the L2 distance since user representations generated by the same user encoder are in the same embedding space. We empirically set the margin $\Delta_{\text{co}} = 1$.

Contrast between Interest and Items. The above objective considers the user representation side solely, and we further capitalize on the target item y_t, which also enhances the user representation learning. Formally, given the L2-normalized representation of the target item $\tilde{\mathbf{y}}$ and user representation $\tilde{\mathbf{x}}$, we have:

$$\mathcal{L}_{ii} = \sum_{m=1}^{M} 1 - \tilde{\mathbf{x}}^{+,m} \cdot \tilde{\mathbf{y}} + \sum_{n=1}^{N} \max\left(0, \tilde{\mathbf{x}}^{-,n} \cdot \tilde{\mathbf{y}} - \Delta_{\text{ii}}\right), \quad (10)$$

This objective also has the advantage of preventing the user encoder from learning trivial representations for counterfactual user sequences. We set the margin $\Delta_{\text{ii}} = 0.5$ in the experiment. Finally, the loss for training the whole framework can be written as:

$$\mathcal{L}_{cause} = \mathcal{L}_{matching} + \lambda_1 \mathcal{L}_{co} + \lambda_2 \mathcal{L}_{ii}. \quad (11)$$

During testing/serving, only the backbone model that generates the user representation is needed. The identification of indispensable/dispensable concepts and the counterfactual transformation processes are disregarded. Noteworthy, the computation of proposal scores which depends on the target item does not belong to the backbone model and is not required during testing.

4 EXPERIMENTS

We conduct experiments on real-world public datasets and mainly aim to answer the following three research questions:

- **RQ1**: How does CauseRec perform compared to the base model and various SOTA sequential recommenders?
- **RQ2**: How do the proposed building blocks and different hyper-parameter settings affect CauseRec?
- **RQ3**: How do user representations benefit from modeling the counterfactual world and contrastive representation learning?

4.1 Experimental Setup

To demonstrate the generalization capability on learning users' representations of the proposed CauseRec architecture, we employ an

Table 1: Statistics of the Datasets.

Dataset	#Users	#Items	#Interactions	#Density
Amazon Books	459, 133	313, 966	8, 898, 041	0.00063
Yelp	31, 668	38, 048	1, 561, 406	0.00130
Gowalla	52, 643	91, 599	2, 984, 108	0.00084

evaluation framework [5, 35, 41] where models should confront unseen user behavior sequences. Specifically, the users of each dataset are split into training/validation/test subset by the proportion of 8 : 1 : 1. For training sequential recommenders, we incorporate a commonly used setting by viewing each item in the behavior sequence as a potential target item and using behaviors that happen before the target item to generate the user's representation, as defined in Section 3.1. For evaluation, only users in the validation/test set are considered, and we choose to generate users' representations on the first 80% behaviors, which are unseen during training. Such a framework can help justify whether models can learn accurate and robust user representations that can generalize well. We mainly focus on the *matching* phase of recommendation and accordingly choose the datasets, comparison methods, and evaluation metrics.

Datasets We consider three challenging recommendation datasets, of which the statistics are shown in Table 1.

- **Amazon Books.** We take Books category from the product review datasets provided by [44], for evaluation. For each user, we keep at most 20 behaviors that are chronologically ordered.
- **Yelp2018.** Yelp challenge (2018 edition) releases the review data for small businesses (*e.g.*, restaurants). We view these businesses as items and use a 10-core setting [20, 65] where each item/user has at least ten interactions.
- **Gowalla.** A widely used check-in dataset [34] from the Gowalla platform. Similarly, we use the 10-core setting [19].

Comparison Methods We mainly consider sequential recommenders for comparison since models are required to confront unseen behaviors for each user. Therefore, factorization-based and graph-based methods are not considered. The compared state-of-the-art models are listed as the following:

- **POP.** A naive baseline that always recommends items with the most number of interactions.
- **YouTube DNN [8].** A successful industrial recommender that generates one user's representation by pooling the embeddings of historically interacted items.
- **GRU4Rec [21].** An early attempt to introduce recurrent neural networks into recommendation.
- **MIND [32].** The first framework that extracts multiple interest vectors for one user based on the capsule network.
- **ComiRec-DR [5].** A recently proposed SOTA framework following MIND to extract diverse interests using dynamic routing and incorporate a controllable aggregation module to balance recommendation diversity and accuracy.
- **ComiRec-SA [5].** ComiRec-SA differs from ComiRec-DR by using self-attention to model interests.

Evaluation Metrics We employ three broadly used numerical criteria for the matching phase, *i.e.*, *Recall*, *Normalized Discounted Cumulative Gain* (NDCG), and *Hit Rate*. We report metrics computed on the top 20/50 recommended candidates. Higher values indicate better performance for all metrics.

Implementation Details We use Adam [30] for optimization with learning rate of 0.003/0.005 for Books/Yelp and Gowalla, $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 1{\times}10^{-8}$, weight decay of $1{\times}1e{-}5$. We train CauseRec-Item for (maximum) 10 epochs and CauseRec-Interest/CauseRec-H for (maximum) 30 epochs with mini-batch size 1024. All models are with embedding size 64. We set hyper-parameters $\lambda_1 = \lambda_2 = 1$ and do not tune them with bells and whistles. As illustrated in Section 3.2, the item encoder is a plain embedding lookup matrix, and the user encoder is a three-layer perceptron with hidden size 256. We set $N = 8$, $M = 1$, $r_{rep} = 0.5$ for CauseRec-Item/-Interest and $N = 16$ $M = 2$ for CauseRec-H to accommodate transformation on two levels, as illustrated in Section 3.3.4. We set $K = 20$ for CauseRec-Interest/-H.

4.2 Performance Analysis (RQ1)

The comparison results of CauseRec with SOTA sequential recommenders are listed in Table 2. We report three architectures of CauseRec including CauseRec-Item (CauseItem), CauseRec-Interest (CauseIn), and CauseRec-Hierarchical (CauseH), as described in Section 3.3.4. In a nutshell, we observe a clear improvement of these architectures over various comparison methods and across three different metrics. Notably, CauseRec-H improves the previous SOTA ComiRec-SA/DR by +.0299 (relatively 22.1%) concerning NDCG@50 on the Amazon Books dataset and +.0179 (relatively 8.64%) concerning Recall@20 on the Gowalla dataset. Among the comparison methods, ComiRec mostly yields the best performance by modeling multiple interests for a given user. However, only modeling the noisy historical behaviors might result in diverse but noisy interests that may not accurately represent users, finally leading to inferior results. GRU4Rec achieves comparably good results with ComiRec on the Gowalla dataset. GRU4Rec can effectively model the sequential dependency between items in the behavior sequence. However, it might be more likely to suffer from the noises due to the strict step-by-step encoding process. In contrast, CauseRec architectures confront the noises within users' behaviors by pushing the user representation away from counterfactually negative user representations and pulling it closer to counterfactually positive user representations. Besides, these results demonstrate the generalization capability of CauseRec on confronting out-of-distribution user sequences by modeling the counterfactual world.

Among three CauseRec architectures, CauseRec-Item is a model-agnostic and non-intrusive design, which means it can be applied to any other sequential recommender without any modification on the original user encoder, and solely functions in the training stage without sacrificing inference efficiency. CauseRec-Interest constructs interest-level concepts by grouping items that may belong to a certain interest (*e.g.*, `chocolate` and `cake` belong to `sweets`) into one holistic concept. Compared to CauseRec-Item, CauseRec-Interest has the advantage of reducing concept redundancy and modeling higher-order relationships between items, and thus improving CauseRec-Item. To combine the merits of CauseRec-Interest and CauseRec-Item, CauseRec-Hierarchical considers both

Table 2: Comparison results of three CauseRec architectures with SOTA sequential recommenders designed for the matching phase. CauseItem/CauseIn/CauseH stand for CauseRec -Item/-Interest/-Hierarchical, respectively. The symbol $*$ indicates the improvements over the strongest baseline (underlined) are statistically significant ($p < 0.05$) with one-sample t-tests.

Datasets	Metric	POP	Y-DNN	GRU4Rec	MIND	ComiSA	ComiDR	CauseItem	CauseIn	CauseH	Improv.
Books	R@20	0.0137	0.0457	0.0406	0.0486	0.0549	0.0531	0.0582	0.0593	**0.0623***	13.5%
	R@50	0.0240	0.0731	0.0650	0.0764	0.0847	0.0811	0.1001	0.0993	**0.1018***	20.2%
	NDCG@20	0.0226	0.0767	0.0680	0.0793	0.0899	0.0918	0.0985	0.1006	**0.1051***	14.5%
	NDCG@50	0.0394	0.1208	0.1037	0.1223	0.1356	0.1352	0.1628	0.1619	**0.1655***	22.1%
	HR@20	0.0302	0.1029	0.0894	0.1062	0.1140	0.1201	0.1280	0.1303	**0.1370***	14.1%
	HR@50	0.0523	0.1589	0.1370	0.1610	0.1720	0.1758	0.2078	0.2062	**0.2113***	20.2%
Yelp	R@20	0.0016	0.0506	0.0454	0.044	0.0534	0.0472	0.0570	0.0580	**0.0591***	10.7%
	R@50	0.003	0.1048	0.0937	0.0943	0.1101	0.0935	0.1163	0.1175	**0.1182***	7.36%
	NDCG@20	0.0065	0.1582	0.1447	0.1414	0.1728	0.1453	0.1812	0.1806	**0.1830***	5.90%
	NDCG@50	0.0129	0.2887	0.2673	0.2699	0.3025	0.2612	0.3179	0.3180	**0.3210***	6.12%
	HR@20	0.0152	0.3015	0.2826	0.2681	0.3249	0.2775	0.3416	0.3391	**0.3426***	5.45%
	HR@50	0.0268	0.5131	0.4853	0.4866	0.5324	0.4629	0.5583	0.5576	**0.5605***	5.28%
Gowalla	R@20	0.0028	0.1127	0.1273	0.1218	0.1277	0.1153	0.1315	0.1355	**0.1359***	6.42%
	R@50	0.0054	0.1926	0.2043	0.2049	0.2072	0.1831	0.2238	0.2204	**0.2251***	8.64%
	NDCG@20	0.0073	0.2378	0.2803	0.2565	0.2736	0.2534	0.2747	0.2825	**0.2842***	1.39%
	NDCG@50	0.0135	0.3638	0.4002	0.3888	0.4019	0.3621	0.4123	0.4113	**0.4221***	5.03%
	HR@20	0.0104	0.3443	0.3814	0.3627	0.3838	0.3429	0.3918	0.3995	**0.4042***	5.32%
	HR@50	0.0224	0.5010	0.5251	0.5301	0.5288	0.5355	0.5596	0.5553	**0.5697***	6.39%

interest-level and item-level concepts in counterfactual transformation. CauseRec-H achieves the best results, which shows that counterfactual transformation on item-level concepts still yields some unique advantages, such as modeling fine-grained preferences. For example, people might not generally like all `sweets` and prefer `cake` to `chocolate`.

4.2.1 Ablation Studies. We are interested in the CauseRec-Item architecture due to its strengths of being: easy to implement (model-agnostic), efficient in serving (non-intrusive), and effective. To obtain a better understanding of different building blocks in CauseRec-Item, we consider surgically removing some components and construct the following architectures. The results on Yelp and Gowalla datasets are shown in Table 3.

w.o. (without) $\mathcal{L}_{co}$. This means we do not consider the contrast between the counterfactual and the observation. The performance drop compared to CauseRec-Item indicates that pushing counterfactually negative user representation away and pulling counterfactually positive user representation closer can potentially help the learned observational user representation to trust more on indispensable items (accurate) and to be immune from dispensable items (robust).

w.o. $\mathcal{L}_{ii}$. $\mathcal{L}_{ii}$ is defined in Equation 10. Removing $\mathcal{L}_{ii}$ means we do not consider the contrast between counterfactual user representations and positive/negative target items. According to Table 3, we observe a clear performance drop compared to CauseRec-Item. Furthermore, eliminating $\mathcal{L}_{ii}$ yields poorer performance than eliminating $\mathcal{L}_{co}$. We attribute this to that, $\mathcal{L}_{ii}$ prevents the user encoder from yielding trivial representations for counterfactual user sequences (item-level and interest-level) by contrasting counterfactual user representations with target items representations. In other words, $\mathcal{L}_{co}$ with possibly trivial counterfactually user representations (without $\mathcal{L}_{ii}$) might hurt the effectiveness $\mathcal{L}_{co}$.

Pos **Only.** CauseRec-Item with only counterfactual transformations on dispensable items and counterfactually positive user representation is denoted as *Pos*. This architecture disregards $\mathcal{L}_{co}$ and term $\sum_{n=1}^{N} \max\left(0, \tilde{\mathbf{x}}^{-,n} \cdot \tilde{\mathbf{y}}_t - \Delta_{\text{margin}}\right)$ in $\mathcal{L}_{ii}$. Not surprisingly, *Pos* Only architecture achieves inferior results compared with w.o. $\mathcal{L}_{co}$ architecture, demonstrating the merits of counterfactually negative user representations. Still, this architecture improves the base model, demonstrating the effectiveness of contrasting counterfactual user representations with target items for recommendation.

Neg **Only.** CauseRec-Item with only counterfactual transformations on indispensable items is denoted as *Neg*. We observe similar results to the *Pos* Only architecture. This again verifies the effectiveness of the contrastive user representation learning framework by modeling the counterfactual distribution. Compared to *Pos* Only architecture, *Neg* Only architecture achieves inferior results. This phenomenon might indicate that making user representation trust more on indispensable concepts can be potentially more important than eliminating the effect of indispensable concepts (*i.e.*, noisy items in CauseRec-Item) on user representation learning. This is reasonable in the sense that the former process might potentially make the user in the embedding space away from all other items, including the dispensable items that are replaced during counterfactual transformation.

Base Model A naive matching baseline described in Section 3.2. All other Architectures yield improvement over the Base Model.

Table 3: Ablation studies by constructing different architectures. We progressively ablate key components in CauseRec-Item, which is a model-agnostic and non-intrusive design.

Model	Yelp						Gowalla					
	Metrics@20			Metrics@50			Metrics@20			Metrics@50		
	Recall	NDCG	Hit Rate	Recall	NDCG	Hit Rate	Recall	NDCG	Hit Rate	Recall	NDCG	Hit Rate
CauseRec-Item	0.0570	0.1812	0.3416	0.1163	0.3179	0.5583	0.1315	0.2747	0.3918	0.2238	0.4123	0.5596
w.o. $\mathcal{L}_{co}$	0.0563	0.1794	0.3350	0.1169	0.3174	0.5507	0.1286	0.2719	0.3865	0.2153	0.4055	0.5522
w.o. $\mathcal{L}_{ii}$	0.0518	0.1679	0.3113	0.1067	0.2983	0.5267	0.1256	0.2668	0.3758	0.2110	0.3968	0.5445
Pos Only	0.0518	0.1662	0.3101	0.1047	0.2911	0.5115	0.1256	0.2693	0.3851	0.2090	0.3945	0.5398
Neg Only	0.0494	0.1633	0.3044	0.1024	0.2909	0.5137	0.1246	0.2608	0.3754	0.2053	0.3925	0.5368
Base Model	0.0444	0.1444	0.2722	0.0934	0.2650	0.4692	0.1208	0.2569	0.3670	0.2000	0.3811	0.5218

4.2.2 Analysis on the number of counterfactual user representations. We ablate the number of counterfactually positive/negative user representations, *i.e.*, M and N, as defined in Section 3.3.4. As shown in Table 4, we can observe that 1) increasing the number (from $N = M = 1$ to $N = M = 8$) not necessarily leads to better performance. 2) particularly increasing the number of counterfactually negative user representations (from $N = M = 1$ to $N = 8, M = 1$) can be useful. This result is reasonable in the sense that such representations can be interpreted as hard negatives since each of the corresponding counterfactual user sequences contains dispensable items staying the same as the original user sequence, and hard negatives are known to be helpful. 3) particularly increasing the number of counterfactually positive user representations (from $N = M = 1$ to $N = 1, M = 8$) may introduce noises since each of them contains some randomly sampled items that can be falsely interpreted as "positive". Therefore one counterfactually positive user representation can be enough for learning accurate user embeddings.

4.2.3 Analysis on the Replace Ratio. We are interested in how the replacement ratio, *i.e.*, r_{rep}, in counterfactual transformation of CauseRec-Item affects the model performance. Table 5 shows the results by varying r_{rep}. The best result is achieved with $r_{rep} = 0.4/0.5$ for the Yelp dataset and $r_{rep} = 0.5$ for the Gowalla dataset. Either too small or too large r_{rep} will lead to sub-optimal results. We attribute this phenomenon to that small r_{rep} will affect the capability of counterfactual learning and large r_{rep} will introduce more noises brought by randomly sampled items for replacement. r_{rep} makes a tradeoff between these two impacts.

4.2.4 Analysis on the number of interest-level concepts. Here we take an analysis on the number of interest-level concepts particularly for CauseRec-Interest architecture. Specifically, we ablate K as described in Equation 6. As shown in Table 6, we observe a performance improvement with K increasing (*e.g.*, $4 \rightarrow 10$, $10 \rightarrow 20$). Each interest-level concept can be more coarse-grained when K becomes smaller and more find-grained when K becomes larger. It can be hard to classify a coarse-grained concept as indispensable or dispensable. It may lead to also coarse-grained or even inaccurate transformations afterward, and thus hurting the performance. Too large K (*e.g.*, 30) may bring redundancies and noises (more random concepts introduced with a fixed replace ratio) to the framework and eventually leads to inferior results.

Table 4: Performance analysis on the number of counterfactually positive/negative user representations in CauseRec-Item, denoted in M/N.

Model	Yelp			Gowalla		
	R@50	N@50	H@50	R@50	N@50	H@50
$N = 1, M = 1$	0.111	0.310	0.541	0.219	**0.413**	0.559
$N = 4, M = 4$	0.113	0.308	0.542	0.210	0.394	0.541
$N = 8, M = 8$	0.106	0.298	0.524	0.204	0.381	0.521
$N = 8, M = 1$	**0.116**	**0.318**	**0.558**	**0.224**	0.412	**0.560**
$N = 1, M = 8$	0.096	0.273	0.483	0.185	0.361	0.498

Table 5: Performance analysis on the replace rate r_{rep} in counterfactual transformation for CauseRec-Item.

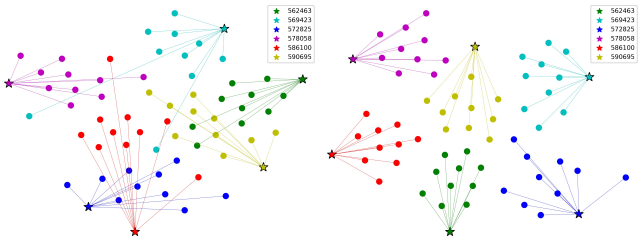
Model	Yelp			Gowalla		
	R@50	N@50	H@50	R@50	N@50	H@50
$r_{rep} = 0.2$	0.115	**0.318**	0.552	0.209	0.389	0.531
$r_{rep} = 0.4$	**0.117**	**0.318**	0.556	0.209	0.401	0.543
$r_{rep} = 0.5$	0.116	**0.318**	**0.558**	**0.224**	**0.412**	**0.560**
$r_{rep} = 0.6$	0.113	0.307	0.537	0.210	0.397	0.537
$r_{rep} = 0.8$	0.106	0.296	0.520	0.211	0.399	0.544

Table 6: Analysis on the number of constructed interest concepts K for CauseRec-Interest.

Model	Yelp			Gowalla		
	R@50	N@50	H@50	R@50	N@50	H@50
$K = 4$	0.100	0.28	0.501	0.206	0.390	0.531
$K = 10$	0.111	0.305	0.536	0.215	0.403	0.546
$K = 20$	**0.118**	**0.318**	**0.558**	**0.220**	**0.411**	**0.555**
$K = 30$	0.117	0.311	0.547	0.219	0.406	0.547

4.3 Case Study (RQ3)

To understand how the learned user representations benefit from the CauseRec framework, we plot randomly selected six users from the Amazon books dataset. We also plot ten corresponding items sampled from the test set for each user. Specifically, we perform



(a) Base Model (b) CauseRec

Figure 3: Visualization of randomly sampled users (shown as stars) with their interacted items (shown as points of the same color) from the Amazon Books dataset. We perform the t-SNE transformation on the representations learned by the base model (left) and CauseRec (right).

t-SNE transformation on the user/item representations learned by the base model (as shown in Figure 3a) and CauseRec-Item (as shown in Figure 3b). The connectivities of users and test items in the embedding space can help reflect whether the model learns accurate and robust user representations. From Figure 3b, we observe that users with their corresponding test items easily form clusters and show small intra-cluster distances and large inter-cluster distances. By jointly comparing the same users (*e.g.*, 590695, and 586100) in Figures 3a and 3b, we can see that CauseRec-Item helps the user encoder learn representations that are closer to their corresponding test items. These results qualitatively demonstrate the effectiveness of CauseRec on learning accurate and robust user representations.

We also present a recommendation result from the Amazon Books test datasets in Figure 4. We list the historical behaviors, the top five books recommended by the base model and CauseRec-Item, and books interacted by the corresponding user in the test set. We mainly visualize the books' covers and categories for better clarity. We note that the side information is generally not considered in training matching models (both the base model and CauseRec). As shown in Figure 4, we observe that CauseRec yields more consistent recommendation results to the books in the test set. Supposing historical behaviors consist of noisy ones, and behaviors in the test accurately reflect users' interest for the current state, CauseRec successfully captures users' interests, *i.e.*, Children's Books, and Literature&Fictions. In contrast, the base model is more likely to be affected by noisy behaviors that appear only a few times, such as the Biographies&Memories, and Education&Reference. These results further demonstrate that CauseRec can learn accurate and robust user representations that are less distracted by noisy behaviors.

5 CONCLUSION AND FUTURE WORK

In this work, we propose to model the counterfactual data distribution to confront the sparsity and noise nature of observed user interactions in recommender systems. The proposed CauseRec conditionally sample counterfactually positive and negative user sequences with transformations on the dispensable/indispensable concepts. We propose multiple structures (-item, -interest, -hierarchical) to confront both fine-grained item-level concepts and abstract interest-level concepts. Several contrastive objectives are devised to

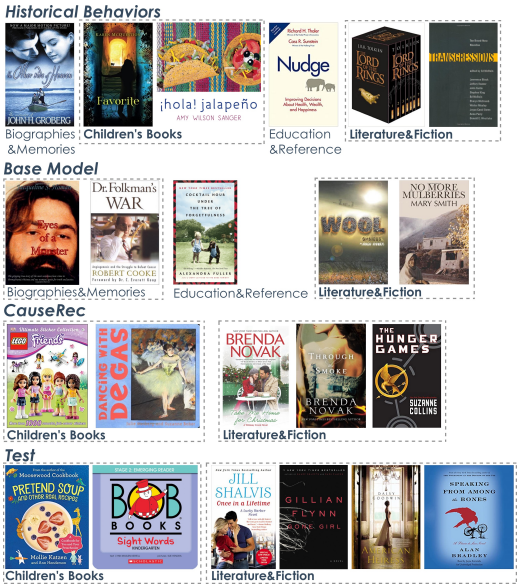


Figure 4: Case study by visualizing a real-world sample from the Amazon Books testing set. We mainly show the books' covers and categories for clarity.

contrast the counterfactual with the observational to learn accurate and robust user representations. Among several proposed architectures, CauseRec-Item has the advantage of being non-intrusive, *i.e.*, solely functioning at training while not affecting serving efficiency. With a naive matching baseline, CauseRec achieves a considerable improvement over it and SOTA sequential matching recommenders. Extensive experiments help to justify the strengths of CauseRec as being both simple in design and effective in performance.

This work can be viewed as an initiative to exploit the joint power of constative learning and counterfactual thinking for recommendation. We believe that such a simple and effective idea can be inspirational to future developments, especially in model-agnostic and non-intrusive designs. CauseRec-Item is compatible with various user encoders within most existing sequential recommenders. We choose a naive baseline to better demonstrate the effectiveness of this work, and we plan to explore its strengths in more models. Another future direction is to whether more effective solutions of identifying indispensable/dispensable concepts exist, including both the computation of concept scores and the determination of indispensable or dispensable for each concept based on the scores. Lastly, we will explore the strengths of CauseRec for the ranking phase of recommendation. Counterfactual transformations designed with various auxiliary features and complex model architectures will open up new research possibilities.

6 ACKNOWLEDGMENTS

The work is supported by the National Key R&D Program of China (No. 2020YFC0832500), NSFC (61625107, 61836002, 62072397), Zhejiang Natural Science Foundation (LR19F020006), and Fundamental Research Funds for the Central Universities (2020QNA5024).

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank.. In *SIGIR*.

[2] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating Position Bias without Intrusive Interventions.. In *WSDM*.

[3] Yoshua Bengio and Jean-Sébastien Senecal. 2008. Adaptive Importance Sampling to Accelerate Training of a Neural Probabilistic Language Model. *IEEE Trans. Neural Networks* (2008).

[4] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation.. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018.*

[5] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation.. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020.*

[6] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks.. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018.*

[7] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.*

[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations.. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016.*

[9] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. 2019. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering* (2019).

[10] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–30.

[11] Fuli Feng, Weiran Huang, Xiangnan He, Xin Xin, Qifan Wang, and Tat-Seng Chua. 2021. Should Graph Convolution Trust Neighbors? A Simple Causal Inference Method. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[12] Hao-Zhe Feng, Kezhi Kong, Minghao Chen, Tianye Zhang, Minfeng Zhu, and Wei Chen. 2020. SHOT-VAE: Semi-supervised Deep Generative Models With Label-aware ELBO Approximations. *arXiv preprint arXiv:2011.10684* (2020).

[13] Hao-Zhe Feng, Zhaoyang You, Minghao Chen, Tianye Zhang, Minfeng Zhu, Fei Wu, Chao Wu, and Wei Chen. 2020. KD3A: Unsupervised Multi-Source Decentralized Domain Adaptation via Knowledge Distillation. *arXiv preprint arXiv:2011.09757* (2020).

[14] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized Ranking Metric Embedding for Next New POI Recommendation.. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015.*

[15] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. 2013. Personalized news recommendation with context trees.. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013.*

[16] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010.*

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning.. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020.*

[18] Ruining He, Chen Fang, Zhaowen Wang, and Julian J. McAuley. 2016. Vista: A Visually, Socially, and Temporally-aware Model for Artistic Recommendation.. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016.*

[19] Ruining He and Julian J. McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback.. In *AAAI.*

[20] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation.. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020.*

[21] 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.*

[22] Balázs Hidasi and Domonkos Tikk. 2016. General factorization framework for context-aware recommendations. *Data Min. Knowl. Discov.* (2016).

[23] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks.. In *SIGIR.*

[24] Rolf Jagerman, Harrie Oosterhuis, and Maarten de Rijke. 2019. To Model or to Intervene: A Comparison of Counterfactual and Online Learning to Rank from User Interactions.. In *SIGIR.*

[25] Sébastien Jean, KyungHyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation.. In *ACL.*

[26] Weike Jin, Zhou Zhao, Pengcheng Zhang, Jieming Zhu, Xiuqiang He, and Yueting Zhuang. 2021. Hierarchical Cross-Modal Graph Consistency Learning for Video-Text Retrieval. In *SIGIR.*

[27] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).

[28] Nathan Kallus, Aahlad Manas Puli, and Uri Shalit. 2018. Removing Hidden Confounding by Experimental Grounding.. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada.*

[29] 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.*

[30] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization.. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

[31] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks.. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.*

[32] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall.. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.*

[33] Yiyang Li, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Content Recommendation by Noise Contrastive Transfer Learning of Feature Representation.. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017.*

[34] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling User Exposure in Recommendation.. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016.*

[35] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering.. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018.*

[36] Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Li Jiwei, and Fei Wu. 2021. BertGCN: Transductive Text Classification by Combining GCN and BERT. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics: Findings.*

[37] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A General Knowledge Distillation Framework for Counterfactual Recommendation via Uniform Data.. 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.*

[38] David C. Liu, Stephanie Rogers, Raymond Shiau, Dmitry Kislyuk, Kevin C. Ma, Zhigang Zhong, Jenny Liu, and Yushi Jing. 2017. Related Pins at Pinterest: The Evolution of a Real-World Recommender System.. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017.*

[39] Hongyu Lu, Min Zhang, and Shaoping Ma. 2018. Between Clicks and Satisfaction: Study on Multi-Phase User Preferences and Satisfaction for Online News Reading.. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.*

[40] Yujie Lu, Shengyu Zhang, Yingxuan Huang, Luyao Wang, Xinyao Yu, Zhou Zhao, and Fei Wu. 2020. Future-Aware Diverse Trends Framework for Recommendation. *CoRR* (2020).

[41] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation.. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.*

[42] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. Pi-Net: A Parallel Information-sharing Network for Shared-account Cross-domain Sequential Recommendations.. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.*

[43] Jarana Manotumruksa and Emine Yilmaz. 2020. Sequential-based Adversarial Optimisation for Personalised Top-N Item 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.*

[44] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes.. In *Proceedings*

of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015.

[45] Michael P. O'Mahony, Neil J. Hurley, and Guenole C. M. Silvestre. 2006. Detecting noise in recommender system databases.. In *Proceedings of the 11th International Conference on Intelligent User Interfaces, IUI 2006, Sydney, Australia, January 29 - February 1, 2006.*

[46] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training.. In *KDD*.

[47] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks.. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017.*

[48] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. 2020. Sequential Recommendation with Self-Attentive Multi-Adversarial Network.. In *SIGIR*.

[49] 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 2010, Raleigh, North Carolina, USA, April 26-30, 2010.*

[50] Nir Rosenfeld, Yishay Mansour, and Elad Yom-Tov. 2017. Predicting Counterfactuals from Large Historical Data and Small Randomized Trials.. In *Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017.*

[51] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules.. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA.*

[52] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation.. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016.*

[53] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks.. In *NIPS*.

[54] 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 *CIKM*.

[55] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding.. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018.*

[56] 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 *NIPS*.

[57] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make It a Chorus: Knowledge- and Time-aware Item Modeling for Sequential 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.*

[58] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item Recommendation with Sequential Hypergraphs.. 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.*

[59] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, ShaoZhang Niu, and Jimmy Huang. 2020. KERL: A Knowledge-Guided Reinforcement Learning Model for Sequential Recommendation.. In *SIGIR*.

[60] 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.*

[61] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-Based Transactional Context Embedding for Next-Item Recommendation.. In *AAAI*.

[62] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2020. "Click" Is Not Equal to "Like": Counterfactual Recommendation for Mitigating Clickbait Issue. *CoRR* (2020).

[63] Wen Wang, Wei Zhang, Jun Rao, Zhijie Qiu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Group-Aware Long- and Short-Term Graph Representation Learning for Sequential Group 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.*

[64] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search.. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018.*

[65] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering.. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.*

[66] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent Recommender Networks.. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017.*

[67] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks.. In *AAAI*.

[68] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination.. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018.*

[69] Xu Xie, Fei Sun, Zhaoyang Liu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive Pre-training for Sequential Recommendation. *CoRR* (2020).

[70] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge J. Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback.. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018.*

[71] Wenwen Ye, Shuaiqiang Wang, Xu Chen, Xuepeng Wang, Zheng Qin, and Dawei Yin. 2020. Time Matters: Sequential Recommendation with Complex Temporal Information.. 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.*

[72] 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.*

[73] Bo-Wen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhenhua Dong, and Chih-Jen Lin. 2019. Improving Ad Click Prediction by Considering Non-displayed Events.. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.*

[74] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and 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.*

[75] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation.. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019.*

[76] Ningyu Zhang, Xiang Chen, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, and Huajun Chen. 2021. Document-level Relation Extraction as Semantic Segmentation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI 2021.*

[77] Ningyu Zhang, Shumin Deng, Xu Cheng, Zhang Yichi Chen, Xi and, Wei Zhang, and Huajun Chen. 2021. Drop Redundant, Shrink Irrelevant: Selective Knowledge Injection for Language Pretraining. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI 2021.*

[78] Shengyu Zhang, Tan Jiang, Tan Wang, Kun Kuang, Zhou Zhao, Jianke Zhu, Jin Yu, Hongxia Yang, and Fei Wu. 2020. DeVLBert: Learning Deconfounded Visio-Linguistic Representations.. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event Seattle, WA, USA, October 12-16, 2020.*

[79] Shengyu Zhang, Ziqi Tan, Zhou Zhao, Jin Yu, Kun Kuang, Tan Jiang, Jingren Zhou, Hongxia Yang, and Fei Wu. 2020. Comprehensive Information Integration Modeling Framework for Video Titling.. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020.*

[80] Zhu Zhang, Zhou Zhao, Zhijie Lin, Xiuqiang He, et al. 2020. Counterfactual Contrastive Learning for Weakly-Supervised Vision-Language Grounding. *Advances in Neural Information Processing Systems* 33 (2020), 18123–18134.

[81] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. 2020. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10668–10677.

[82] Lei Zheng, Ziwei Fan, Chun-Ta Lu, Jiawei Zhang, and Philip S. Yu. 2019. Gated Spectral Units: Modeling Co-evolving Patterns for Sequential Recommendation.. In *SIGIR*.

[83] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2020. Contrastive Learning for Debiased Candidate Generation at Scale. *CoRR* (2020).

[84] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization.. In *CIKM*.