User Modeling and User-Adapted Interaction
https://doi.org/10.1007/s11257-022-09332-z

Improving session-based recommendation with contrastive learning

Wenxin Tai[1] · Tian Lan[1] · Zufeng Wu[1] · Pengyu Wang[1] · Yixiang Wang[1] · Fan Zhou[1]

Received: 3 October 2020 / Accepted in revised form: 29 May 2022

Abstract

Session-based recommendation, which aims to predict the next item given anonymous behavior sequences of users, is critical in modern recommender systems. While prior works have made efforts to improve recommendation performance, two challenges remain unsolved. First, existing learning methodologies rely on mining sequential patterns within the individual session and use the next item as the supervised signal, which may not effectively capture the correlations among interactions. Second, previous solutions are also limited in learning the mixed dependencies inside flexibly ordered sessions, i.e., sequential dependencies among ordered interactions and non-sequential dependencies among unordered ones. This work presents a novel session recommender algorithm by distilling knowledge and supervision signals from sessions in a contrastive manner. We propose position-aware importance extraction module with contrastive learning, which utilizes the intrinsic dependencies to discover extra knowledge and augment the ability of information distillation. Besides, we introduce a bi-directional matching algorithm with contrastive loss to capture potential patterns through maximizing the mutual information between current interaction and historical behaviors. Moreover, we introduce a simple yet effective learnable position-

✉ Fan Zhou
fan.zhou@uestc.edu.cn

Wenxin Tai
wxtai@std.uestc.edu.cn

Tian Lan
lantian1029@uestc.edu.cn

Zufeng Wu
wuzufeng@uestc.edu.cn

Pengyu Wang
p.y.wang@std.uestc.cn

Yixiang Wang
yxwang@std.uestc.edu.cn

1 University of Electronic Science and Technology of China, Chengdu, China

Published online: 21 June 2022

coding mechanism with self-attention-based importance extraction to flexibly learn user browsing patterns. Extensive experiments conducted on two real-world datasets demonstrate that our proposed algorithm enhances the recommendation performance over existing state-of-the-art approaches.

Keywords Session-based recommendation · Self-supervised learning · Position-aware embedding · Contrastive learning · Long-term preference

1 Introduction

Recommender systems (RS) have become indispensable tools supporting online users by providing potential items of interest. Most existing recommender systems assume that the user profile and previous activities are recorded constantly. However, in many cases, user identification may be unknown, and only user behavior history during an ongoing session is available due to various reasons such as privacy issues. Under such realistic circumstances, conventional recommendation methods that rely on adequate user-item interactions have problems in yielding accurate recommendations (Rendle et al. 2010, 2012). Hence, session-based recommender systems (SBRS) have emerged with increasing attention in recent years.

In the last decade, session-based recommendation has received considerable attention from both industry and academia. GRU4REC (Hidasi et al. 2016a) first uses a recurrent neural network (RNN) to capture sequential user behaviors and learn the recommendation model. NARM (Li et al. 2017) proposes to understand users' primary purpose with the attention mechanism. STAMP (Liu et al. 2018) captures users' general and current interests using simple MLP networks and an attentive network that makes the model more efficient. SR-GNN (Wu et al. 2019) firstly uses a graph neural network (GNN) to reconstruct item representation and then leverages an attention mechanism to obtain the user preference vectors. Recently, SR-IEM (Pan et al. 2020b) points out that judging the user's interest preference solely by the mixture of items or the last click behavior is ineffective. To address this problem, they propose an importance extraction module (IEM) that applies a modified self-attention mechanism to extract the importance of each item in an ongoing session. Compared with previous competitive neural models, SR-IEM achieves considerable improvements in terms of HitRate and mean reciprocal rank. Another line of works aims at enriching the current session representation through cross-session information fusion (Wang et al. 2019; Qiu et al. 2020).

Despite the promising results achieved by the aforementioned approaches, two key challenges prevent them from generating satisfactory results. First, existing deep-learning-based recommendation approaches train their models through minimizing the gap between the user demands and the predictive distribution, which has been proved insufficient for distilling crucial signals from user's behaviors (Wang et al. 2017; Zhang et al. 2020). This may happen due to the overemphasis of final output results, which may not fully explore the rich contexts and transition patterns inherent in a session (Yang et al. 2019).

Second, existing methods capture sequential transition regularities of user behaviors, assuming the strict sequential orders of user behaviors, which, however, is not always held in practice. For example, Markov-chain-based methods (Hidasi et al. 2016a, b; Li et al. 2017; Ren et al. 2019; Xu et al. 2020) follow the rigidly ordered assumption over intra-session item transition, which may not capture nonlinear interactions between users and items. Gated recurrent unit (GRU) has been utilized to model nonlinear sequential correlations between past and future behaviors (Hidasi et al. 2016a, b), and has achieved improvement over linear models. Recently, attention-based methods (Liu et al. 2018; Tang et al. 2018), convolutional neural network (CNN)-based methods (Yuan et al. 2019) and Graph-based methods (Wu et al. 2019; Xu et al. 2019) relax the exact orders assumption within the sessions, which makes them more robust in complex scenarios. Nevertheless, a flexibly-ordered session is neither totally unordered nor precisely ordered, i.e., some parts of the session are ordered while others are not. Therefore, completely ignoring or over-emphasizing the item transitions regularities will limit the representation ability to exist deep neural networks (Zhang et al. 2019). Thus, how to effectively model the continuous interactive process is critical for improving sequential recommendation performance.

To address above challenges, we propose a novel session-based recommendation method **PIE-CL** (Position-aware Importance Extraction module with Contrastive Learning), which is inspired by recent advances in self-supervised learning (SSL). SSL has been widely studied in computer vision (CV) (Tian et al. 2020; Khosla et al. 2020; He et al. 2020; Chen et al. 2020) and natural language processing (NLP) (Mikolov et al. 2013; Vaswani et al. 2017; Devlin et al. 2019), and shown comparable performance as supervised approaches in a range of image recognition and NLP tasks (Liu et al. 2020). PIE-CL is a deep neural network-based recommendation model, equipped with two specific designs: (i) A contrastive learning algorithm that distills supervision signals from the session data itself and enhances the representation through mutual information maximization; and (ii) A combination of affinity matrix-based self-attention and learnable position encoding mechanisms, which enables the model to learn potential location patterns adaptively. Figure 1 compares the commonly used predictive learning and the self-supervised contrastive learning proposed in this paper. The contributions of this paper are threefold:

– We propose a generic and effective data-driven session-based recommendation model PIE-CL that combines the affinity matrix-based self-attention algorithm and learnable position encoding mechanism in an end-to-end manner. Our model is general and can be easily extended to various sequential recommendation scenarios.
– We present a mutual information-based self-supervised multi-task learning paradigm to better capture user's intrinsic preferences. To our knowledge, our model is the first attempt addressing session-recommendation with contrastive learning objectives.
– We prove the rationality of the proposed contrastive-based algorithm by the theoretical analysis. Meanwhile, we demonstrate the effectiveness of the proposed method with extensive experimental evaluations.

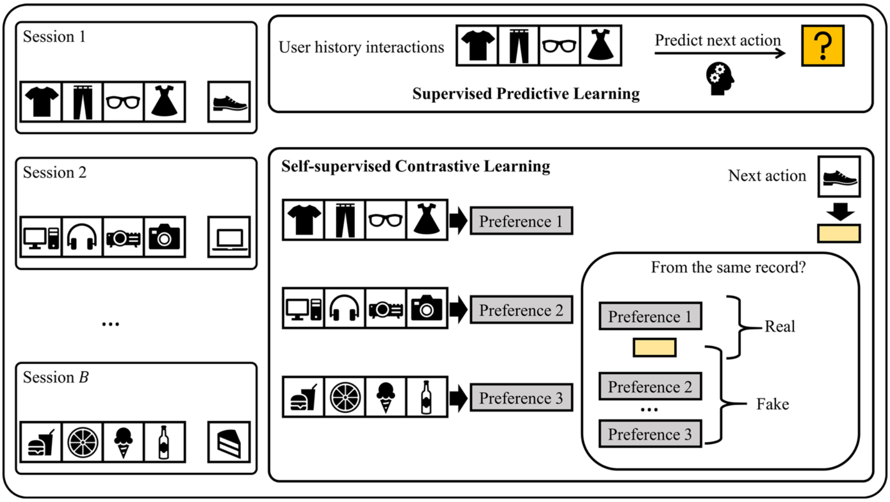


Fig. 1 Supervised predictive learning versus self-supervised contrastive learning. Supervised predictive learning is widely adopted in existing works which uses single matching losses such as cross-entropy. In our self-supervised learning method, the intrinsic correlations between items are learned by the signals from the session data itself using the contrastive loss

The rest of this work is organized as follows. In Sect. 2, we review the related work and the state-of-the-art models for the session-based recommendation, followed by the preliminaries of problem definition and mutual information maximization in Sect. 3. In Sect. 4, we present the proposed method as well as the details of the model training. We experimentally evaluate the performance of PIE-CL and show the results in Sect. 5. Finally, we conclude this paper and point out the directions for future work in Sect. 6.

2 Related work

Most of the existing SBRS solutions take the items and actions as the input while ignoring their attribute information (Liu et al. 2018; Twardowski 2016). In various SBRS, the input is usually formalized as a session context conditioned on which the recommendation is performed. According to the difference of specific tasks, session-based recommendations include next session recommendation, next partial-session recommendation, and next interaction recommendation. Wang et al. (2021) Especially in the E-Commerce domain, improvement in these topics can bring benefits for both business and customers. In the next session recommendation, the output is a list of complementary interactions (items) to recommend the next session (Wang et al. 2021), while in the next partial-session recommendation, the output is to predict a series of items to complete the current session. This work focus on the third challenge (a.k.a next item prediction/recommendation), which receives the most attention in relevant research topics. Given the known part (e.g., happened interactions) of a session, the goal of the next interaction recommendation is to recommend the next possible interaction in the current session by using the historical record of user actions. It is usually simplified to predict the next item to interact with, e.g., item/product clicks so far.

This section first reviews the conventional session-recommendation approaches and the recent advances in intra-session recommendation algorithms. Then, cross-session methods, as well as researches related to flexibly ordered information mining, are discussed. Also, we briefly introduce the latest contrastive learning studies.

2.1 Conventional session-based recommendation algorithms

The most popular SBRS approach is collaborative filtering (CF), which models the user interest based on the whole history. For example, matrix factorization (MF) (Koren et al. 2009) is a general approach. The basic objective is to factorize a user-item rating matrix into two low-rank matrices, each of which represents the latent factors of users or items. However, information about long-term individual preferences in session-based recommender systems is not avaliable (He et al. 2016). As a result, clustering-based algorithms and Markov-chain-based algorithms have become the mainstream in SBRS. Sarwar et al. (2001) prove that the heuristics-based nearest neighbor(kNN) scheme is simple but effective for SBRS. The proposed approach calculates a score for each candidate interaction based on the similarity scores calculated on the co-occurrence of items in the training set. Adomavicius and Tuzhilin (2005) propose an improved version called the session-aware collaborative method that recommends items based on popularity in the current session. Compared with item-KNN, session-KNN considers the whole session context rather than the current item in the session context and thus can capture more relevant information for more accurate recommendations (Hariri et al. 2015). Jannach and Ludewig (2017) present a session-based kNN method that incorporates heuristics to sample suitable neighbors.

Mobasher et al. (2002) propose one of the earliest session-based approaches based on frequent pattern mining for the recommendation of Web pages to visit. In principle, they study different sequential patterns for recommendation and find that contiguous sequential patterns are more suitable for sequential prediction tasks than general sequential patterns. Shani et al. (2005) present an MDP (Markov Decision Process) approach for session-based recommendations in e-commerce and demonstrate its value from a business perspective. Rendle et al. (2012) propose a hybrid model FPMC, which models sequential behavior between every two adjacent clicks and provides a more accurate prediction for each sequence. The main drawback of the Markov-chain-based models is that they combine past components independently that restricts the prediction performance.

2.2 Intra-session-based deep learning algorithms

Deep neural networks have been proven to be very effective in modeling sequential data in recent years (LeCun et al. 2015; He et al. 2016). Inspired by recent advances in natural language processing (Socher et al. 2011; Sutskever et al. 2014), some deep-learning models have been developed and achieved impressive improvements for SBRS (Hidasi et al. 2016a, b; Hu et al. 2017; Li et al. 2017). Numerous researchers use deep learning

(DL) methods to improve SBR results. Hidasi et al. (2016a) propose GRU4REC, the first attempt applying RNN networks to solve the SBR problem. This model takes users' historical behavior as input and makes predictions of the next item relying on the sequential modeling capability of the RNNs. Li et al. (2017) stack GRU as an encoder to extract sequential interactions and leverages the attention mechanism to obtain the users' preference. To alleviate the bias of the data, Liu et al. (2018) replace the recurrent encoder with an attention layer. Ren et al. (2019) argue that people may repeat their actions in an ongoing session. Thus, they propose a repeat-explore mechanism with an encoder–decoder structure for a repeat consumption phenomenon. Wu et al. (2019) apply a gated graph network (Li et al. 2016) as the item feature encoder to extract item embeddings from a session graph, which are then fed into an attentive network to generalize the final representation for the next item prediction. Yu et al. (2020) propose a target-aware attention network, which adaptively learns different user interests concerning target items, thereby letting the learned interest representation vectors vary with different target items. Furthermore, some works (Pan et al. 2020b; Yu et al. 2020; Qiu et al. 2020) argue that determining the attention weight by only relying on the last clicked item is unreasonable. Inspired by the success of Transformers (Vaswani et al. 2017; Devlin et al. 2019) in NLP tasks, Pan et al. (2020b) use a modified self-attention mechanism based on the affinity matrix to estimate the item importance in a session.

2.3 Cross-session-based algorithms

Despite the effectiveness of the approaches mentioned above, the intra-session-based deep learning models only focus on the item transitional relations within a single session, which is insufficient for information extraction if a session is short. Some studies (Wang et al. 2015; Quadrana et al. 2017; Wu and Yan 2017; Bai et al. 2018) attempt to leverage the cross-session information by the link of sessions that belong to the same user. In 2017, Wu and Yan (2017) design a session-aware method to pre-train the session representations by incorporating different kinds of user search behaviors such as clicks and views. Meanwhile, Quadrana et al. (2017) apply a recurrent architecture to aggregate information from user's historical records. In 2019, Bai et al. (2018) adopts an attention mechanism to combine different sessions. However, the above algorithms are inapplicable to the anonymous sessions without user identification. In the SBRS area, Wang et al. (2019) design a collaborative SBR machine that incorporates the neighbor sessions as auxiliary information. Qiu et al. (2020) construct a broadly connected session graph used to learn item contextual representations with the preservation of implicit correlations between items across different sessions.

2.4 Modeling flexibly ordered session

A flexibly ordered session is neither totally unordered nor ordered, i.e., some parts of the session are ordered while others are not (Tang et al. 2018). For example, a tourist generates a session of check-ins at the airport, hotel, shopping center, and bar successively. In this session, the airport, hotel, and bar are sequentially dependent, while the shopping center is randomly checked without any order. Therefore, the complex dependencies inside the flexibly ordered sessions must be carefully considered and explicitly learned for accurate recommendation (Wang et al. 2021).

Some of the existing session-based recommendation models rely on the rigid order assumption of item transitional relationships, i.e., artificial-decay-factors-based methods (Campos et al. 2014; Garg et al. 2019) or RNN-based approaches (Hidasi et al. 2016a; Li et al. 2017; Ren et al. 2019; Wang et al. 2019). Although the aforementioned Markov-based models have achieved satisfactory performance, they only produce suboptimal results when facing non-chronological sequences due to their over-emphasis on the associations between adjacent actions. Compared with sequential dependencies, most of the co-occurrence-based dependencies among interactions are collective dependencies (Tang et al. 2018; Yuan et al. 2019). Some purely attention-based methods (Liu et al. 2018; Pan et al. 2020b) are widely used because of their computational convenience. Nevertheless, completely ignoring the possible sequential information is not a proper way for a better recommendation system (Niranjan et al. 2010). Therefore, it is of great importance that an SBR model could effectively recognize the user's behavior in a flexible-ordered session without completely ignoring or overemphasizing the item transition regularities. Table 1 summarizes the main methods for SBRS that are most closely related to this work.

2.5 Contrastive learning

Recently, contrastive learning has achieved remarkable successes in various applications, such as speech modeling (Oord et al. 2018), image processing (Hjelm et al. 2019; Zhou et al. 2021; He et al. 2020; Baxter 2000; Bollmann and Søgaard 2016), and graph learning (Velickovic at al. 2019; Zhou et al. 2021, c). Different from the supervised predictive paradigms, contrastive learning-based models try to distinguish positive and negative samples. The models can ignore the general feature representation of the shallow layers to achieve this goal but retain the distinguishable features. Kong et al. (2020) bridge the gap between contrastive learning and mutual information and explain that contrastive learning works from the perspective of maximizing mutual information between the anchor sample and the positive one(s). Arora et al. (2019) propose to accommodate positive and negative samples in various forms by constructing blocks with theoretical guarantees and performances. Tian et al. (2020) argue that the anchor samples and the positive samples should be paired with each other to design a bidirectional and symmetrical contrastive framework. Apart from using contrastive learning for pre-training as described above, Yang et al. (2019) prove that the use of contrastive learning in the training process can yield remarkable results. Zhou et al. (2020) introduce contrastive learning into recommender systems to alleviate exposure bias in deep candidate generation (Covington et al. 2016), while self-supervised trip recommendation has been presented in a recent work (Zhou et al. 2021b).

Table 1 Summary of the main studies in session-based recommendation

Reference	Technique	Attention	Markov	Ordered	Un-ordered	Intra-session	Cross-session	Graph	Contrastive
Sarwar et al. (2001)	Item-based collaborative				✔		✔		
Mobasher et al. (2002)	Markov-based		✔	✔			✔		
Adomavicius and Tuzhilin (2005)	Session-based collaborative				✔		✔		
Shani et al. (2005)	Markov decision process				✔		✔		
Rendle et al. (2010)	Matrix factorization		✔		✔		✔		
Davidson et al. (2010)	Item-based KNN				✔		✔		
Hidasi et al. (2016a)	Recurrent neural network		✔	✔		✔			
Benson et al. (2016)	Sequential repeat consumption			✔			✔		
Jannach and Ludewig (2017)	Session-based KNN				✔		✔		
Li et al. (2017)	Recurrent, Attention	✔	✔	✔		✔			
Liu et al. (2018)	Attention	✔			✔	✔			
Garg et al. (2019)	Position-aware, Collaborative				✔		✔		
Wu et al. (2019)	Graph neural network	✔			✔	✔		✔	
Xu et al. (2019)	Graph, Attention	✔			✔	✔		✔	
Ren et al. (2019)	Repeat pattern	✔	✔	✔		✔			
Wang et al. (2019)	Memory bank	✔	✔	✔		✔	✔		
Yu et al. (2020)	Target-aware, Graph	✔			✔	✔		✔	
Qiu et al. (2020)	Board Graph	✔			✔		✔	✔	
Pan et al. (2020b)	Affinity matrix	✔			✔	✔			
Current	Position-aware, Contrastive	✔		✔	✔	✔	✔		✔

3 Preliminaries

In this section, we start with formally defining the session-based recommendation problem and then provide the necessary background w.r.t. mutual information and contrastive learning.

3.1 Problem formulation

Session-based recommendation aims to predict which item a user will click next based on the user's sequential behaviors in the current session without accessing to the user's long-term historical data. Formally, given a session $\mathcal{S} = \{x_1, x_2, ..., x_t\}$ that consists of t items that the user has interacted with (e.g., clicked and purchased), the goal of session-based recommendation is to predict the next interaction at time step $t+1$ from an item embedding set of n items $\mathcal{V} = \{v_1, v_2, ...v_n\}$.

Table 11 (cf. Appendix) summarizes the frequently used notations.

3.2 Mutual information maximization

Mutual information (MI) is a Shannon entropy-based measurement of random variable dependencies (Belghazi et al. 2018). Given two variables X and Y, the mutual information is

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{1}$$

3.3 Self-supervised learning by contrasting samples

The main idea of self-supervised learning (SSL) is to pre-train a model on a large amount of data using the self-supervised signals, e.g., measuring the distance between positive and negative samples without label supervision. The SSL does not require intensive handcrafted labels that may greatly improve the model's generalizability and robustness. Specifically, SSL considers three main types of data including the *anchor*, *positive*, and *negative* samples. The distance between the anchor x and a positive sample x^+ should be smaller than the distance between x and a negative sample x^- in the latent space of the learned representations, i.e.,

$$f_\theta(x, x^+) \gg f_\theta(x, x^-), \tag{2}$$

where $f_\theta(\cdot, \cdot)$ is a similarity function (e.g., dot product or cosine similarity). For example, the goal of learning representations for a negative sample is to maximize:

$$\max \left[\frac{f_\theta(x, x^+)}{f_\theta(x, x^+) + f_\theta(x, x^-)} \right], \tag{3}$$

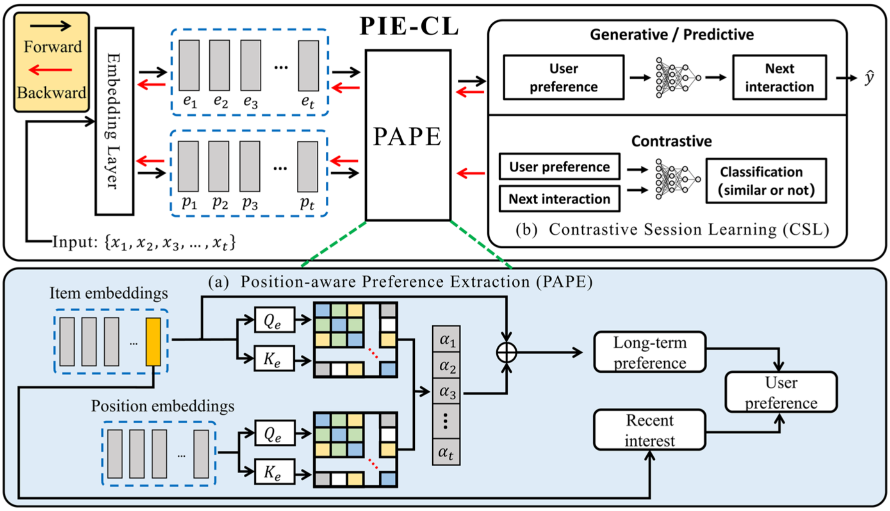


Fig. 2 Architecture of the proposed PIE-CL. The input to the model is $E_e = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_t\}$ and $E_p = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_t\}$, which are converted from the input session $\mathcal{S} = \{x_1, x_2, \ldots, x_t\}$. Position-aware Preference Extraction Module (PAPE) serves as a generator of user's preference. In contrast, Contrastive Session Learning Module (CSL) is applied to distill supervision signals and streghten intrinsic correlations. By compared with the items in $\mathcal{V}$, a recommendation result $\hat{y}$ is finally generated

which has been used for loss functions in many models (Kong et al. 2020; Schmarje et al. 2020), e.g., the widely used InfoNCE (Gutmann and Hyvärinen 2010):

$$\mathcal{L} = -\mathbb{E}_{(x,x^+)}\left[f_\theta(x, x^+) - \log \sum_{x_i \in N_{\text{neg}}} \exp f_\theta((x, x_i)) \right], \tag{4}$$

where N_{neg} denotes the set of negative samples.

4 PIE-CL: architecture and methodology

In this section, we present the details of the proposed model PIE-CL.

4.1 Overview

PIE-CL consists of two main components, i.e., (1) the position-aware preference extraction (PAPE) module, and (2) the contrastive session learning (CSL) module. Figure 2 outlines the architecture of PIE-CL. PAPE aims to learn the user's long-term preference and recent interest based on the position-aware affinity-matrix-based self-attention mechanism. Unlike previous attention-based methods that discard sequential information, PAPE uses a learnable position-coding scheme to learn item-based and position-based affinity matrices separately. Furthermore, we introduce CSL—a self-supervised learning module that distills supervision signals from the session data itself and strengthens the intrinsic correlations among user's behaviors by maximizing the likelihood of distinguishing the potential differences across sessions. In the following, we discuss each component of PIE-CL in detail.

4.2 Position-aware importance extraction

Conventional Markov-chain-based methods (Campos et al. 2014; Hu et al. 2017; Garg et al. 2019) and RNN-based methods (Hidasi et al. 2016a; Li et al. 2017; Ren et al. 2019; Wang et al. 2019) consider user behaviors based on the models' sequential learning ability, which, however, fail to capture the importance of items, and, more importantly, assume perfect sequential relations in a session. In practice, the user behaviors in a session are full of randomness and chaos that may deteriorate the prediction performance of the sequential models, which, therefore. Later, numerous attention-based models (Li et al. 2017; Liu et al. 2018; Ren et al. 2019; Wang et al. 2019) are introduced to explicitly consider the correlation between the historical clicks and the last click. These methods consider and calculate dynamic weights for an ongoing session, but they still heavily rely on the recent clicks to learn users' interests.

Recently, self-attention-based approaches (Zhang et al. 2019; Pan et al. 2020b) achieve state-of-the-art performance in SBRS, due to its effectiveness in determining the importance weights of each historical item. These models are dependent on the self-attention mechanism (Vaswani et al. 2017), but fail to model the transition relations inherent in a session. Taking SR-IEM as an example, it transforms item embeddings $E_e = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_t\}$ into a low-dimensional space via a nonlinear function by computing a affinity matrix $\mathcal{A}$ with a query matrix $\mathbf{Q}$ and a key matrix $\mathbf{K}$. The attention weight of each item can be written as:

$$\begin{aligned}\alpha_i &= \text{softmax}\left(\frac{1}{\sqrt{d}}[\sum_{j=1, j\neq i}^{t} \mathcal{A}_{ij}\right)\\ &= \text{softmax}\left(\frac{1}{\sqrt{d}}\left[\sum_{j=1, j\neq i}^{t} (\mathbf{Q}\mathbf{K}^T)_{ij}\right]\right)\\ &= \text{softmax}\left(\frac{1}{\sqrt{d}}\sum_{j=1, j\neq i}^{t} [\mathcal{F}\{\mathbf{W}_q\mathbf{E})\mathcal{F}(\mathbf{W}_k\mathbf{E})^T]_{ij}\right)\\ &= \text{softmax}\left(\frac{1}{\sqrt{d}}\{\mathcal{F}(\mathbf{e}_i\mathbf{W}_q)[\sum_{j=1, j\neq i}^{t} \mathcal{F}(\mathbf{e}_j\mathbf{W}_k)^T]\}\right),\end{aligned} \tag{5}$$

where $\mathcal{F}$ denotes the activation function, $\mathbf{W}_q$ and $\mathbf{W}_k$ are the learnable projection matrices. As we can see, it does not make use of any sequential information, i.e., it is permutation-invariant.

To preserve the relative position relations, various position embeddings have been introduced into transformer (Vaswani et al. 2017) based models, e.g., BERT (Devlin et al. 2019) and its many variants, to capture the order relationships in the latent space. The weight calculation can be accordingly rewritten as:

$$\alpha_i = \text{softmax}\left(\frac{1}{\sqrt{d}}\{\mathcal{F}((\mathbf{e}_i + \mathbf{p}_i)\mathbf{W}_q)\left[\sum_{j=1, j\neq i}^{t} \mathcal{F}((\mathbf{e}_j + \mathbf{p}_j)\mathbf{W}_k)^T\right]\}\right) \tag{6}$$

which can be further expanded as:

$$\begin{aligned}
\alpha_i &= \text{softmax}\left(\frac{1}{\sqrt{d}}\left\{\mathcal{F}((\mathbf{e}_i + \mathbf{p}_i)\mathbf{W}_q)\left[\sum_{j=1, j\neq i}^{t} \mathcal{F}((\mathbf{e}_j + \mathbf{p}_j)\mathbf{W}_k)^T\right]\right\}\right)\\
&= \text{softmax}\left(\frac{1}{\sqrt{d}}\left\{\mathcal{F}(\mathbf{e}_i\mathbf{W}_q)\left[\sum_{j=1, j\neq i}^{t} \mathcal{F}(\mathbf{e}_j\mathbf{W}_k)^T\right]\right\}\right) +\\
&\quad \text{softmax}\left(\frac{1}{\sqrt{d}}\left\{\mathcal{F}(\mathbf{p}_i\mathbf{W}_q)\left[\sum_{j=1, j\neq i}^{t} \mathcal{F}(\mathbf{p}_j\mathbf{W}_k)^T\right]\right\}\right) +\\
&\quad \text{softmax}\left(\frac{1}{\sqrt{d}}\left\{\mathcal{F}(\mathbf{e}_i\mathbf{W}_q)\left[\sum_{j=1, j\neq i}^{t} \mathcal{F}(\mathbf{p}_j\mathbf{W}_k)^T\right]\right\}\right) +\\
&\quad \text{softmax}\left(\frac{1}{\sqrt{d}}\left\{\mathcal{F}(\mathbf{p}_i\mathbf{W}_q)\left[\sum_{j=1, j\neq i}^{t} \mathcal{F}(\mathbf{e}_j\mathbf{W}_k)^T\right]\right\}\right)
\end{aligned} \tag{7}$$

which shows how the position embedding and item embedding are projected and queried. We can see that there are four terms after the expansion, i.e., *item-to-item*, *item-to-position*, *position-to-item*, and *item-to-item correlations*. From this reformulation, we have several concerns: (1) it can be easily observed that this kind of position-coding is able to consider positional correlations indeed; (2) nevertheless, it also introduces noise such as item-to-position and position-to-item modeling (Ke et al. 2020).

To better consider position embedding and word embedding while eliminating unnecessary noise, we introduce a dual-path position-aware item importance extraction module in our PIE-CL. First, we create a learnable position embedding dictionary and generate corresponding position embeddings $E_p = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_t\}$. Subsequently, we compute the item-based and the position-based affinity matrices separately:

$$\mathcal{A}_e = \frac{\mathcal{F}(\mathbf{W}_{qe}\mathbf{E}_e)\mathcal{F}(\mathbf{W}_{ke}\mathbf{E}_e)^T}{\sqrt{2d}}. \tag{8}$$

$$\mathcal{A}_p = \frac{\mathcal{F}(\mathbf{W}_{qp}\mathbf{E}_p)\mathcal{F}(\mathbf{W}_{kp}\mathbf{E}_p)^T}{\sqrt{2d}}. \tag{9}$$

where $\mathbf{W}_{qe}$, $\mathbf{W}_{ke}$, $\mathbf{W}_{qp}$, and $\mathbf{W}_{qp}$ are learnable projection matrices; $\sqrt{2d}$ is used to retain the magnitude. We use tanh instead of sigmoid as the activation function to obtain a wider representation space due to the regularization constraint. Since the affinity matrix suggests that an item is not important if its corresponding similarity scores related to other items are relatively low, we compute the importance score α_i of each item by combining the affinity-matrix-based attentive method and the proposed new positional encoding mechanism:

$$\begin{aligned}\alpha_i &= \mathrm{softmax}\left(\frac{1}{\sqrt{d}}[\sum_{j=1, j\neq i}^{t} (\mathcal{A}_e + \mathcal{A}_p)_{ij}\right) \\ &= \mathrm{softmax}\left(\frac{1}{\sqrt{d}}\left\{\mathcal{F}(\mathbf{e}_i \mathbf{W}_{qe})\left[\sum_{j=1, j\neq i}^{t} \mathcal{F}(\mathbf{e}_j \mathbf{W}_{ke})^T\right]\right\}\right) + \\ &\quad \mathrm{softmax}\left(\frac{1}{\sqrt{d}}\left\{\mathcal{F}(\mathbf{p}_i \mathbf{W}_{qp})\left[\sum_{j=1, j\neq i}^{t} \mathcal{F}(\mathbf{p}_j \mathbf{W}_{kp})^T\right]\right\}\right) \end{aligned} \tag{10}$$

After generating the item importance weights, we combine the user's long-term preference and recent interest within the session to generate a final session representation as to the user preference. As previous work has shown that the last item in a session can represent a user's recent interest (Li et al. 2017; Liu et al. 2018), we directly take the representation of the last item as the user's recent interest, i.e.,$\mathbf{z}_s$. Considering that items in a session have different degrees of importance, the long-term user's preference $\mathbf{z}_l$ with regard to the current session can be calculated as follows:

$$\mathbf{z}_l = \sum_{i=1}^{t} \alpha_i \mathbf{e}_i. \tag{11}$$

Then, we concatenate the long-term preference and current interest as the final representation:

$$\mathbf{z} = [\mathbf{z}_l; \mathbf{z}_s]. \tag{12}$$

After obtaining the user's preference, we use it to make recommendations by calculating the probabilities of all candidate items. That is, we calculate the score $\hat{z}$ of each item in the item set $\mathcal{V}$ by multiplying the session representation with all item embeddings. The score of a certain item is:

$$\hat{z}_i = \mathbf{z}^T \mathbf{W}_0 \mathbf{v}_i, \tag{13}$$

where $\mathbf{W}_0$ is a projection matrix used to calculate the similarity scores. Here, we apply a softmax layer to normalize the preference scores of candidate items. Finally, the items with the highest scores will be recommended to the user.

$$\hat{y} = \mathrm{softmax}(\hat{z}). \tag{14}$$

The training algorithm of the proposed PIE-CL model is summarized in Algorithm 1.

Algorithm 1: Training PIE-CL.

Input: Sessions $\mathcal{S} = \{x_1, x_2, ..., x_t\}$.
Output: The learned parameters **2** of PIE-CL.

Shuffle training data;
for *each batch in the training data* **do**
  `/* Encoding module. */`
  for $i = 1 \rightarrow t$ **do**
    Calculate the position of the item x_i in a session;
  end
  Feed Imporance Extraction module with $\mathbf{E}_e$ and $\mathbf{E}_p$;
  `/* Importance Extraction Module. */`
  Compute affinity matrix $\mathcal{A}_e$ and $\mathcal{A}_p$ via Eq. (8) and Eq. (9);
  Compute importance weight α_i of each item via Eq. (10);
  Construct user's long-term preference via Eq. (11);
  Generate user's preference representation $\mathbf{z}$ via Eq. (12);
  `/* Parameters Optimization. */`
  Calculate the forward matching cross-entropy loss via Eq. (15);
  Calculate the reverse matching binary InfoNCE loss by Algorithm 2;
  Update **2** by mininizing Eq. (20) .
end

4.3 Contrastive learning

Existing SBRS models either rely on a single objective function (e.g., the cross-entropy loss and KL divergence), or are trained in a supervised learning manner. For example, once the user's preference representation in a session has been generated, existing models use the distribution gap between the predictions and ground truths as the supervision signals:

$$\mathcal{L} = \sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) + \text{Norm}||\mathbf{\Theta}||^2. \tag{15}$$

We dim that there exist some potential connections between the historical records and the next interaction since they are visited by the same user in a short period. It is of great importance if the recommender system can efficiently capture such potential patterns. To this end, we propose a novel contrastive learning paradigm to distill auxiliary supervision signals from the session itself and capture the intrinsic correlations between the historical behaviors and the next interaction. Specifically, we use the future interaction x_{t+1} to match its corresponding history records from a number of distracters. Assume x_{t+1} is the anchor x, the session(s) consisting of the same next item in the current batch will be considered as the positive sample(s), while the remaining are negative samples. Inspired by Arora et al. (2019), we construct two blocks that are used to aggregate positive samples and negative samples in the current batch to obtain more stable positive and negative representations:

$$\mathbf{z}_i^+ = \frac{1}{B^+} \sum_{j=1}^{B} \mathbb{I}(y_i = y_j) \cdot \mathbf{z}_j, \tag{16}$$

$$\mathbf{z}_i^- = \frac{1}{B^-} \sum_{j=1}^{B} \mathbb{I}(y_i \neq y_j) \cdot \mathbf{z}_j, \tag{17}$$

where y_i is the ground-truth label w.r.t. the future interaction for the i-th session, and z_i is the representation of the session i in current batch:

The binary InfoNCE loss (Khosla et al. 2020) can be used to train the model:

$$\mathcal{L}_{CL}^i = -\log \frac{\exp(\mathbf{x}_{it}^T \mathbf{W}_0 \mathbf{z}^+/\tau)}{\exp(\mathbf{x}_{it}^T \mathbf{W}_0 \mathbf{z}^+/\tau) + \exp(\mathbf{x}_{it}^T \mathbf{W}_0 \mathbf{z}^-/\tau)}, \tag{18}$$

where τ is the temperature hyper-parameter. Note that the binary InfoNCE loss is very similar to the triplet loss (Liu et al. 2020). For simplicity, we use $x \cdot x^+$ to represent $\mathbf{x}_{it}^T \mathbf{W}_0 \mathbf{z}^+$ and $x \cdot x^-$ to represent $\mathbf{x}_{it}^T \mathbf{W}_0 \mathbf{z}^-$:

$$\begin{aligned}
\mathcal{L}_{CL}^i &= -\log \frac{\exp(x \cdot x^+/\tau)}{\exp(x \cdot x^+/\tau) + \exp(x \cdot x^-/\tau)} \\
&= \log(1 + \exp((x \cdot x^+ - x \cdot x^-)/\tau)) \\
&\approx \exp((x \cdot x^+ - x \cdot x^-)/\tau) \\
&\approx 1 + \frac{1}{\tau}(x \cdot x^+ - x \cdot x^-) \\
&= 1 - \frac{1}{2\tau} \cdot (||x - x^+||^2 - ||x - x^-||^2) \\
&\propto ||x - x^+||^2 - ||x - x^-||^2 + 2\tau.
\end{aligned} \tag{19}$$

Equation (19) has the same form as a triplet loss with a margin of $\alpha = 2\tau$. The conclusion is consistent with empirical results in Khosla et al. (2020), He et al. (2020) and Kong et al. (2020). From the perspective of metric learning, the inner product corresponds to a simple metric on the high-dimensional space. Besides, it also highlights the importance of choosing the negative samples appropriately. The pipeline of our proposed contrastive session learning is shown in Fig. 3.

Notably, the gradient of the contrastive loss is subject to $\mathbf{z}^+$ and $\mathbf{z}^-$ (Eq. (16) and Eq. (17)), which in turn depends on the statistics of the samples in the whole batch set. The proposed contrastive learning algorithm has the potential to suppress the spurious gradients from outliers and thus improves the generalization of the model (Kumar et al. 2016).

Algorithm 2 outlines the procedure of contrastive session training.

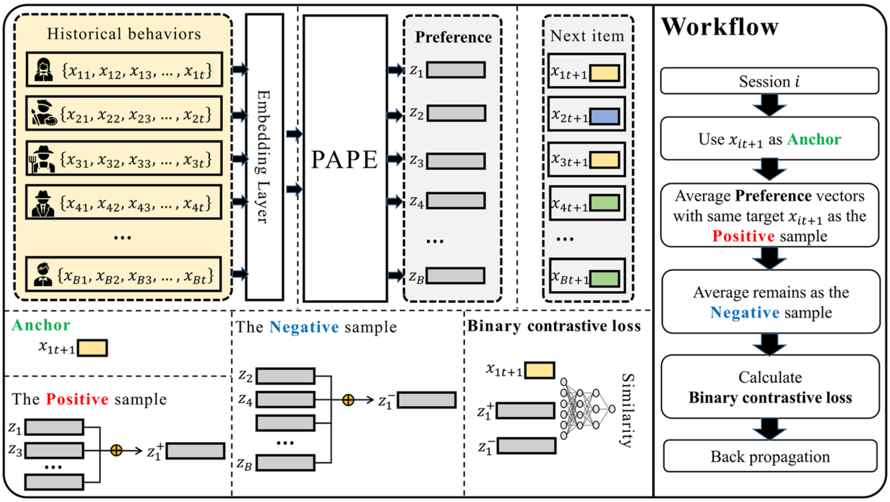


Fig. 3 Illustration of the contrastive session learning

Algorithm 2: Procedure of Contrastive Learning.

Input: Session representation $\mathbf{z}_i$ and corresponding target item embedding $\mathbf{e}_i$
Output: $\mathcal{L}_{CL}$

```
/* Training Steps                                                    */
```
for *each batch* **do**
  for *each record i* **do**
    Prepare ground truth embedding vector as the anchor;
    Average all session representations with the same target as the positive sample $\mathbf{z}^+$ via Eq. (16);
    Average the negative sample $\mathbf{z}^-$ via Eq. (17);
    Calculate the matching scores via $\mathbf{x}_{it}^T\mathbf{W}_0\mathbf{z}^+$ and $\mathbf{x}_{it}^T\mathbf{W}_0\mathbf{z}^-$;
    Obtain contrastive learning-based binary InfoNCE loss via Eq. (18);
  end
  Average the contrastive loss for each session in current batch;
end

To train our model, we combine cross-entropy and contrastive losses through a hyper-parameter λ as the optimization objective to update the model parameters:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \lambda \cdot \mathcal{L}_{CL}, \tag{20}$$

4.4 Complexity analysis

The main overhead of PIE-CL lies in the following three parts: (1) calculation of the attention weights; (2) projection mapping of session representation; and (3) contrastive-based auxiliary classification task. In particular, obtaining the affinity matrix is the core consumption of attention weight calculation, which requires $O(t^2d)$

time complexity—where t is the session length, and d denotes the embedding dimensions. At the last layer of the recommendation network, the mapping projection complexity is $O(nd^2)$. In the self-supervised training of PIE-CL, it involves three operations: (1) generate anchor, core positive, and the negative samples in the current batch; (2) calculate the similarity between the anchor and the positive or negative samples; and (3) backpropagation via proposed contrastive binary InfoNCE loss. In each batch, the three operations result in an additional $O(B^2d)$ complexity. Since $t < d$ and $B < n$, the auxiliary task learning is more efficient than main computations such as attention calculation and projection.

5 Experiments

In this section, we report the results of extensive experimental evaluations on two real-world datasets to verify the performance of the proposed PIE-CL. Specifically, we try to answer following research questions in this section

- **Q1**: How does PIE-CL perform compared with the state-of-the-art session-based recommendation models?
- **Q2**: How do different components in PIE-CL affect the prediction performance?
- **Q3**: Is the proposed contrastive-based auxiliary framework applicable to the other models?
- **Q4**: Can PIE-CL provide reasonable explanations about its prediction behavior?
- **Q5**: How do key hyper-parameters influence PIE-CL's performance?

5.1 Datasets

We evaluate the performance of the proposed model on two benchmark datasets, namely the *Yoochoose* (Yoo) dataset from the RecSys'15 Challenge[1] and the *Diginetica* (Digi) dataset from the CIKM'16 competition[2].

For a fair comparison, following previous work, we filter out all sessions of length 1 and items appearing less than 5 times in two datasets. The remaining 7,981,580 sessions and 37,483 items constitute the Yoochoose dataset, while the Diginetica dataset consists of 204,771 sessions and 43,097 items. Similar to Hidasi et al. (2016a), we generate sub-sessions and corresponding labels by splitting the original record. Following previous works (Pan et al. 2020b), we set the maximum session length L to 10, i.e., for each session, we keep the 10 most recent interactions only. Besides, we use the most recent fractions of the training sequences of Yoochoose, i.e., 1/64 and 1/4, following related works (Li et al. 2017; Liu et al. 2018; Wu et al. 2019). The statistics of datasets are summarized in Table 2.

[1] http://2015.recsyschallenge.com/challege.html.

[2] http://cikm2016.cs.iupui.edu/cikm-cupl.

Table 2 Statistics of the datasets

Statistics	Yoochoose1/64	Yoochoose1/4	Diginetica
Clicks	557,248	8,326,248	982,961
Train sessions	369,859	5,917,745	719,470
Test sessions	55,898	55,898	60,858
Items	16,766	30,470	43,097
Average length	6.16	5.71	5.12

5.2 Baselines

We compare our PIE-CL with the following 14 state-of-the-art baselines, including traditional approaches and deep-learning based models: (i) popularity-based recommendation strategy (i.e., POP and S-POP); (ii) K-nearest neighbor modeling algorithm (i.e., item-KNN); (iii) traditional personalized matrix factorization techniques (i.e., FPMC and BPR-MF); (iv) RNN-based models (i.e., GRU4REC, NARM, and RepeatNet); (v) un-ordered recommendation models (i.e., STAMP and SR-IEM); (vi) Graph-based models (i.e., SR-GNN); (vii) cross-session models (i.e., CSRM).

- **POP** (Sarwar et al. 2001) always recommends the most popular items in the training set.
- **S-POP** (Adomavicius and Tuzhilin 2005) recommends the top-N frequent items based on the occurrence frequency in current session.
- **FPMC** (Rendle et al. 2010) is a hybrid model for next-basket recommendation. To adapt it to session-based recommendation, we ignore the user latent representations when computing recommendation scores.
- **BPR-MF** (Rendle et al. 2012) is a commonly used matrix factorization method. We apply it to session-based recommendation by representing a new session with the average latent factors of items that occurred in the session so far.
- **Item-KNN** (Davidson et al. 2010) recommends items similar to the current item. The similarity is defined as the co-occurrence of two items in sessions.
- **GRU4REC** (Hidasi et al. 2016a) uses RNNs to model user's sequential behaviors for next item preidction.
- **SKNN** (Jannach and Ludewig 2017) is a session-based KNN approaches. The basic idea is to find past sessions that contain the same elements as the ongoing session. The recommendations are then based on selecting items that appeared in the most similar past session.
- **NARM** (Li et al. 2017) extends GRU4REC and improves the capability of session modeling with the attention mechanism.
- **STAMP** (Liu et al. 2018) emphasizes the importance of both user's current and general preference. The attention mechanism is built on top of the embedding of the last click that represents the user's current interests.
- **STAN** (Garg et al. 2019) is a session-based KNN methods. Based on SKNN, STAN additionally takes time and position information into account when predicting the next item.

- **CSRM** (Wang et al. 2019) is a hybrid framework that considers cross-session information. It constructs a dynamic memory bank to explore the neighborhood information for better predicting the intent of the user in current session.
- **SR-GNN** (Wu et al. 2019) transfers an item sequence to a structured graph and uses GNN to represent the session sequences. Based on the session graph, SR-GNN is capable of capturing the transitions of items and generating item embeddings correspondingly, which are difficult to be modeled by conventional sequential methods like MC-based and RNN-based methods.
- **RepeatNet** (Ren et al. 2019) is an encoder–decoder structure with repeat-exploration mechanism to better consider the intent of user behavior
- **SR-IEM** (Pan et al. 2020b) is an importance extraction model that applies a modified self-attention mechanism to extract the importance of each item in an ongoing session.

All baselines are trained with the optimal parameter settings described in the original papers. We report the average results of three-time runs for all methods.

5.3 Evaluation metrics

Following Ludewig et al. (2021), we report the results on HitRate (HR), Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), and Coverage (COV).

HitRate@K is calculated over top-K items, i.e., the proportion of the correct recommended items in the K previous positions in a ranking list:

$$HR@K = \frac{n_{\text{hit}}}{S}, \tag{21}$$

where S is the size of the test set, and n_{hit} is the number of times that the desired item appears in the top K position.

MRR@K corresponds to the average reciprocal ranks over the desired items. If the rank of item v_t is less than or equal to K, its MRR value will be set to zero; otherwise, it would be retained and used for average calculation:

$$MRR@K = \frac{1}{S} \sum_{t \in V, Rank(t) < K} \frac{1}{Rank(t)} \tag{22}$$

NDCG@K is another ranking-dependent metric of ranking quality in information retrieval tasks. Due to the particularity of next interaction recommendation, the original function can be abbreviated as:

$$NDCG@K = \frac{1}{S} \sum_{s \in S} \sum_{i=1}^{K} \frac{2^{rel(i)} - 1}{\log_2 (1 + i)}, \tag{23}$$

where $rel(i)$ is a binary function that returns 1 if the recommended item in the candidate list, or else it will return 0.

COV@K refers to the proportion of items that can be recommended to the total items. Specifically, we calculate the number of different items in the recommendation list and record the ratio to the total number of items in the dataset.

5.4 Experimental settings

For all baselines, the parameters are the same as those reported in the original papers. For our model, the dimensions of the item embeddings and attentions are set to 200 and 100, respectively. We use Adam optimizer with an initial learning rate of 10^{-3} and a decay factor of 0.1 for every three epochs. The batch size is set to 128, and L2 regularization is used to avoid overfitting by setting $L2 = 10^{-5}$. The hyper-parameters λ and τ are 10 and 1, respectively. Note that the data preprocessing of baselines in the contrastive ablation study (Sec. 5.6) is slightly different. For example, it is inappropriate to truncate the sessions for RepeatNet and SR-GNN, since they require longer sessions to capture the periodicity and high-order neighbors, respectively. Besides, one of the main contributions of this paper is to provide an extensible contrastive learning-based paradigm, which should not be affected by parameter settings during contrastive ablation studies. Hence, when evaluating the generalizability of the proposed contrastive learning (Sec. 5.7), we keep the same configurations as SR-GNN for STAMP and RepeatNet, i.e., using the original session length without clipping; while for SR-IEM and PIE-CL, we use the default parameter settings as described in SR-IEM. To help reproduce the results of our model, we have made our PIE-CL code publicly available[3].

5.5 Performance comparison (Q1)

Table 3 shows the experimental results when $K = 20$. From the results, we have following important observations.

First, although GRU4Rec that rely on RNN for modeling user behaviors performs better than heuristic and matrix factorization-based methods such as POP and FPMC, it may not fully capture user intentions especially when the user's goal is unclear. This happens because this kind of sequential models may be confused by the user's session behavior that are full with click uncertainty. The attention mechanism can discriminate the importance of sequential behaviors, which can alleviate the uncertainty of user behaviors and help capture users' real preferences. Therefore, compared with GRU4REC, approaches that are based on attention mechanisms such as NARM and STAMP achieve higher recommendation performance.

Second, SR-GNN establishes complex transformation relationships among items through learning graph representations, which indeed improves session recommendation performance. However, when modeling the user's long-term interest, the importance of each item is only determined by the correlation with the item last clicked. This scheme is "myopic" since it ignores the long-term dependencies and may easily "fooled" by the unintentional click, which also explains why its perfor-

[3] https://github.com/judiebig/PIE-CL .

Table 3 Performance comparisons ($K = 20$) on Yoochoose and Diginetica

Method	Yoochoose1/64		Yoochoose1/4		Diginetica	
	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR@20
POP	6.71	1.65	1.33	0.30	0.89	0.20
S-POP	30.44	18.35	27.08	17.75	21.06	13.68
FPMC	45.62	15.01	–	–	26.53	6.95
BPR-MF	31.31	12.08	3.40	1.57	5.24	1.89
Item-KNN	51.60	21.81	52.31	21.70	35.75	11.57
SKNN	63.77	25.22	62.13	24.82	48.06	16.95
STAN	69.45	28.74	70.07	28.89	50.97	18.48
GRU4REC	60.64	22.89	59.58	22.62	29.45	8.33
NARM	68.32	28.63	69.75	29.30	49.70	16.17
STAMP	68.74	29.67	70.44	30.00	45.64	14.32
CSRM	69.85	29.71	70.63	29.48	51.69	16.92
SR-GNN	70.57	30.94	71.36	31.89	50.73	17.59
RepeatNet	70.79	31.40	70.71	31.03	47.79	17.66
SR-IEM	70.75	31.45	71.35	31.63	51.00	16.87
PIE-CL	**71.25***	**31.70***	**71.83***	**31.90***	**52.20***	**17.70***

*Statistically significant improvement of PIE-CL over the best baseline according to a paired t-test ($p < 0.01$)

mance is lower than that self-attention mechanism-based methods SR-IEM. CSRM, as a cross-session-based approach, achieves appealing results owing to its unique external memory bank mechanism that can exploit extra information from surrounding neighbor sessions. However, the extra storage mechanism increases the memory and computational overhead, which is also common in GNN-based approaches modeling inter-session dependencies (Qiu et al. 2020).

Third, our PIE-CL consistently outperforms all baseline approaches on two datasets. Compared with the best baseline models SR-IEM, PIE-CL consists of an addition of learnable position-coding mechanism, enabling the self-attention mechanism to consider flexible ordered information. Furthermore, PIE-CL is a contrastive learning approach that can learn to ignore the shallow features and noise clicks while capturing those deep-seated and well-distinguished ones through the comparison between different sessions.

Tables 4, 5, 6 and 7 summarize the HitRate@K and MRR@K results of our PIE-CL and several strong baselines varying with the values of K – * indicates statistical significance under the t-test ($p < 0.01$), and more numerical results can be found in the Appendix. We can see that the advantages of our model become more evident with increasing the values of K. Taking Diginetica as an example, when K=1, PIE-CL achieves 0.53% performance gains over SR-IEM in terms of both HitRate and MRR, whereas when K=15, the improvements are 1.12% and 0.78%, respectively.

Although the improvement achieved by PIE-CL is slight for a smaller K, the performance gain is non-trivial. The characteristics of datasets limit the performance

Table 4 HR@K results on Yoochoose1/64, $K = [1, 3, 5, 10, 15]$

Method	Yoochoose1/64				
	HR@1	HR@3	HR@5	HR@10	HR@15
STAMP	17.22	36.78	46.75	59.51	66.03
RepeatNet	18.31	37.04	47.32	60.01	66.77
SR-GNN	17.78	37.32	47.56	60.28	66.22
SR-IEM	18.49	37.62	47.10	60.32	66.71
PIE-CL	**18.56***	**37.92***	**48.09***	**60.90***	**67.27***

Table 5 MRR@K results on Yoochoose1/64, $K = [1, 3, 5, 10, 15]$

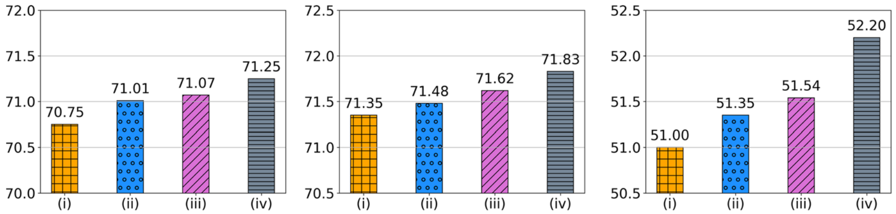
Method	Yoochoose1/64				
	MRR@1	MRR@3	MRR@5	MRR@10	MRR@15
STAMP	17.22	25.65	27.93	29.65	30.17
RepeatNet	18.31	26.39	28.74	30.47	30.99
SR-GNN	17.78	26.17	28.51	30.22	30.72
SR-IEM	18.49	26.72	29.03	30.75	31.25
PIE-CL	**18.56***	**26.89***	**29.29***	**30.95***	**31.45***

Table 6 HR@K results on Diginetica, $K = [1, 3, 5, 10, 15]$

Method	Diginetica				
	HR@1	HR@3	HR@5	HR@10	HR@15
STAMP	8.56	19.62	26.75	38.21	45.79
RepeatNet	7.71	18.20	25.32	37.07	44.61
SR-GNN	8.78	19.18	26.17	37.57	45.07
SR-IEM	8.29	18.78	25.96	37.68	45.40
PIE-CL	**8.82***	**19.76***	**27.07***	**38.87***	**46.52***

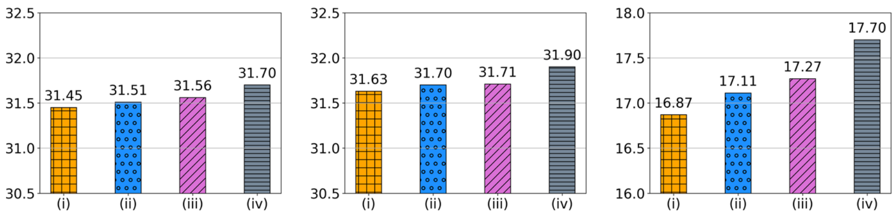
Table 7 MRR@K results on Diginetica, $K = [1, 3, 5, 10, 15]$

Method	Diginetica				
	MRR@1	MRR@3	MRR@5	MRR@10	MRR@15
STAMP	8.56	13.39	15.01	16.53	17.12
RepeatNet	7.71	12.15	13.77	15.32	15.92
SR-GNN	8.78	13.21	14.80	16.31	16.90
SR-IEM	8.29	12.77	14.40	15.95	16.56
PIE-CL	**8.82***	**13.51***	**15.17***	**16.74***	**17.34***



(a) HR@20 on 1/64 **(b)** HR@20 on 1/4 **(c)** HR@20 on Digi

Fig. 4 HR@20 results of ablation study



(a) MRR@20 on 1/64 **(b)** MRR@20 on 1/4 **(c)** MRR@20 on Digi

Fig. 5 MRR@20 results of ablation study. Since the nDCG results are the same trend as MRR, we only report the MRR scores

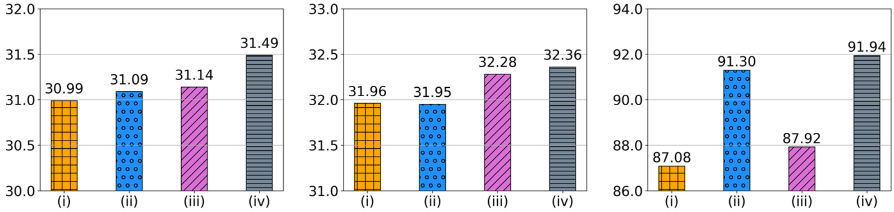
of session-based recommendation models for smaller values of K. For example, the total number of candidates on both Yoochoose and Diginetica datasets is vast (40k-50k items), making it difficult for recommender systems to precisely predict the user's behavior if only one item is recommended—the probability of matching user behavior exactly through random prediction is only 0.0025%.

PIE-CL relies on contrastive learning with reverse matching to extract the items relevant to real user preference. In essential, PIE-CL optimizes item representations by emphasizing the most possible items (positive samples) while diminishing the influence of negative items. Therefore, the top-K items output by PIE-CL is more likely to contain the user's actual action. Therefore, the increase of K expands the scope of CL's influence, making items output by PIE-CL more likely to contain the user's actual action. This also explains why the performance gain of our PIE-CL becomes great when increasing K.

5.6 Ablation study (Q2)

We now investigate the effect of individual components in PIE-CL through an ablation study. Specifically, we implemented three variants of PIE-CL, including: (i) SR-IEM, a basic model that only uses self-attention mechanism, (ii) IE-CL, a variant of PIE-CL without position, (iii) PIE, a variant of PIE-CL without contrastive learning, and (iv) the full PIE-CL model.

Figures 4, 5 and 6 show the detailed performance of the four models in terms of HitRate, MRR and Coverage, respectively. We have the following insights that allow us to understand how individual component helps learn user preferences. First,



(a) Cov@20 on 1/64 **(b)** Cov@20 on 1/4 **(c)** Cov@20 on Digi

Fig. 6 Cov@20 results of ablation study

Table 8 Statistics of the testing data

Dataset	Total item	Item coverage (%)	Number label	Label coverage (%)	Label variance
Yoochoose	37,484	18.01	6175	16.47	292.51
Diginetica	43,098	49.03	19,631	45.54%	10.67

contrastive-learning-based reverse matching boosts the performance regardless of adding the position-coding mechanism or not. Besides, both position-coding mechanism and contrastive-learning-based auxiliary task can improve the results over basic architecture. Importantly, the contributions of position-coding and contrastive learning mechanisms are complementary, which thereby demonstrates the critical importance of both mechanisms.

Interestingly, the Cov@20 results on Diginetica dataset is significantly higher than on Yoochoose, according to a paired t-test ($p < 0.01$). We analyzed the two datasets and found that the model covers 6,175 items in Yoochoose, while the Diginetica test set covers 19,631 items that is three times more (shown in Table. 8). Therefore, compared with Yoochoose, Diginetica is more suitable for evaluating SBRS methods since it has a more comprehensive testing set.

5.7 Generalizability of contrastive learning (Q3)

To further determine the effect and generalizability of our contrastive-learning-based bi-directional matching scheme, we incorporate it into several deep learning-based models, including STAMP, RepeatNet, SR-GNN, and SR-IEM.

Figure 7 shows that the contrastive learning-based auxiliary task improves the performance of all baseline approaches in most cases. This result proves our motivation of improving SBRS performance with contrastive auxiliary task learning that can distill extra signals from the complex user behaviors in a self-supervised learning manner. The only exception is SR-GNN in the Yoochoose dataset, where the performance of SR-GNN drops slightly after adding the proposed contrastive-based auxiliary loss. We conjecture that this phenomenon is caused by data distribution characteristics in Yoochoose, i.e., the imbalance of data distribution and the limitation of evaluating models using biased testing data. That is, the uneven distribution of item frequency may lead to the generalizability issue. For example, the testing data only cover a

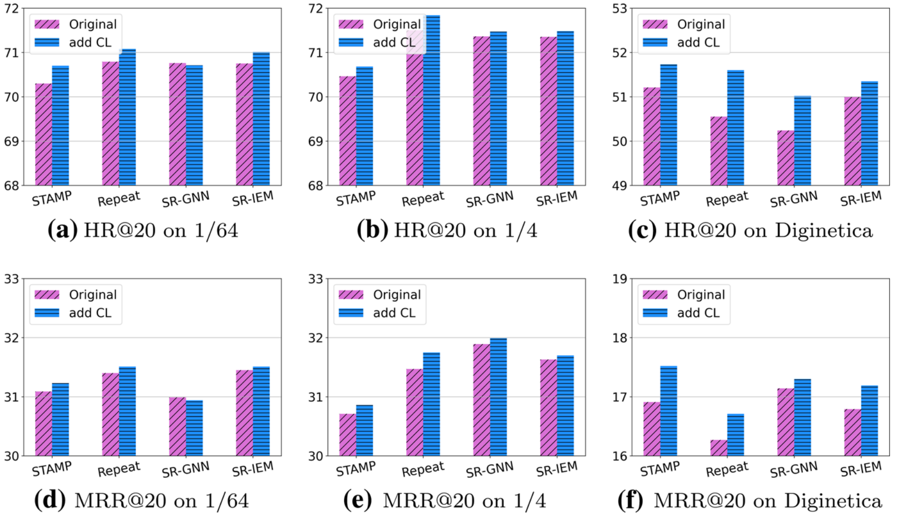


Fig. 7 Effect of contrastive learning in other baselines. (**a**), (**b**), (**c**) show R@20 performance, while (**d**), (**e**), (**f**) show MRR@20 results. Here, we use the same parameters and experimental settings reported in SR-GNN

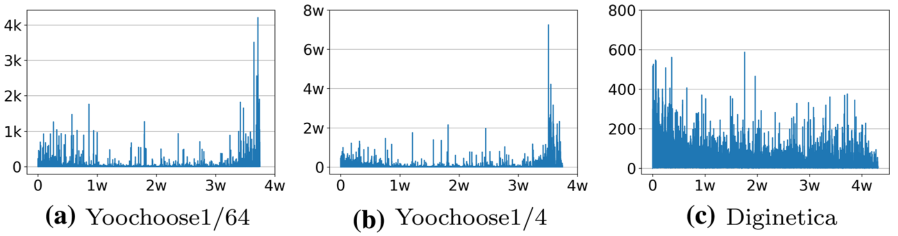
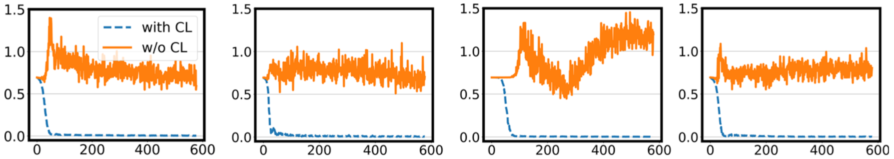


Fig. 8 Illustration of the item frequency distribution. The x-axis represents the item id, and the y-axis represents the count number of each item in the dataset

small portion of items, which would result in poor generalization from training data to testing data. To validate our hypothesis, we plot the item frequency distribution of data illustrated in Fig. 8. Compared to Diginetica, the item frequency distribution in Yoochoose is highly uneven. According to the statics of testing data (Table 8), only 18.01% items occur in the testing data in Yoochoose, which is significantly less than in Diginetica. Besides, as a GNN-based method, SR-GNN has been well trained with the supervised signals in the graph to learn user's transition patterns (Xu et al. 2019; Pan et al. 2020) in the smallest dataset. Therefore, the room left for improvement with contrastive learning is limited. When data distribution becomes more balanced, contrastive learning still shows strong ability to extract auxiliary signals from the data itself.



(a) PIE-CL (b) SR-GNN (c) RepeatNet (d) STAMP

Fig. 9 Binary InfoNCE loss during training on Yoochoose1/64. The x-axis represents the training round, and the y-axis denotes the loss

5.8 Interpretability (Q4)

We provide a full explanation of why our proposed contrastive learning-based auxiliary approach works from three perspectives, i.e., metric learning, regularization constraint, and multi-task learning(MTL)/transfer learning(TL), respectively. From the perspective of metric learning, the matching scores between history sessions and the next interaction that from the same record should be as high as possible compared with those from other records (Tian et al. 2020). The introduction of contrastive-based auxiliary tasks explores the intrinsic session dependencies to discover extra knowledge via providing additional reverse self-supervised objectives. The contrastive-learning-based auxiliary matching and main supervision tasks constitute a bi-directional matching pattern, which provides a principled way to characterize the inherent data correlations while tackling the implicit feedback and weak supervision problems by learning robust representations applicable for the session-based recommendation.

From the perspective of regularization constraint, we believe the additional contrastive loss can enhance model training and prediction accuracy via mutual information-based self-supervised training. To verify this, we draw the training loss of reverse matching before and after using the contrastive auxiliary framework, as shown in Fig. 9.

There are two interesting observations. First, the model does not have the ability of symmetric matching without using contrastive learning, thereby confirming our previous conjecture that previous models cannot capture the distinguishing features effectively. Instead, it learns shallow identical characteristics, making some user's preference representation close to each other in the latent space and increasing the risk of over-fitting. After adding the contrastive constraint, the model tends to capture the personalized and fine-grained characteristic features in learning user preferences, which greatly alleviates the aforementioned problem (Caruana 1993). Second, the contrastive-based reverse matching loss can converge after a few epochs, e.g., less than 100. This is because the contrastive loss serves as a constraint to limit the optimization space of parameters during training, i.e., it provides potential constraints of the parameters.

From the perspective of MTL, MTL is effective in that it offers multiple different perspectives compared with single-task learning (Yang et al. 2019; Zhang et al. 2021). The prediction quality of commonly used multi-task models is often sensitive to the relationships between tasks. It is therefore important to study the trade-offs between task-specific objectives and inter-task relationships. In the proposed architecture, we design an auxiliary module that is strongly related to the original prediction task and use Binary InfoNCE loss that is more robust than triplet loss, as an additional optimization function. The proposed multi-task learning framework for the session-based recommendation can simultaneously learn potential transitions among user's behavior records via a bi-directional matching diagram.

5.9 The order of the user behaviors (Q4)

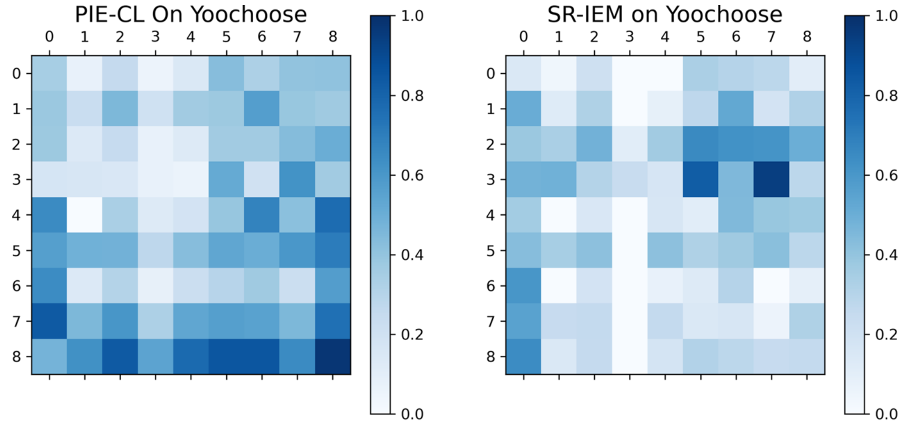
Previous experiments have proven that the position coding mechanism is crucial for SBRS. Now, we will further explore how the position coding works through a series of experiments.

Apart from the superior recommendation performance, another key advantage of PIE-CL lies in its ability to adaptively adjust the importance weights of item correlations by flexibly modeling the orders of user behaviors. Toward this goal, we use case studies to show the interpretability of our model through visualizing the self-attention weights obtained from PIE-CL. In particular, we visualize the affinity matrix to illustrate the explainability of the PIE-CL. Figure 10 shows four heatmaps of self-attention weights from one randomly selected session. Compared with SR-IEM, PIE-CL with a position-coding mechanism obtains the additional temporal information, guiding the model to allocate the attention weight better. In addition, a few consecutive items are related to the next click in the current session, and the most important items often appear at the end of the session (the summation of the i-th row corresponds to the self-attention weight of i-th item). This result verifies that the complex dependencies inside a flexibly ordered session can be carefully considered and precisely learned through our position encoding approach.

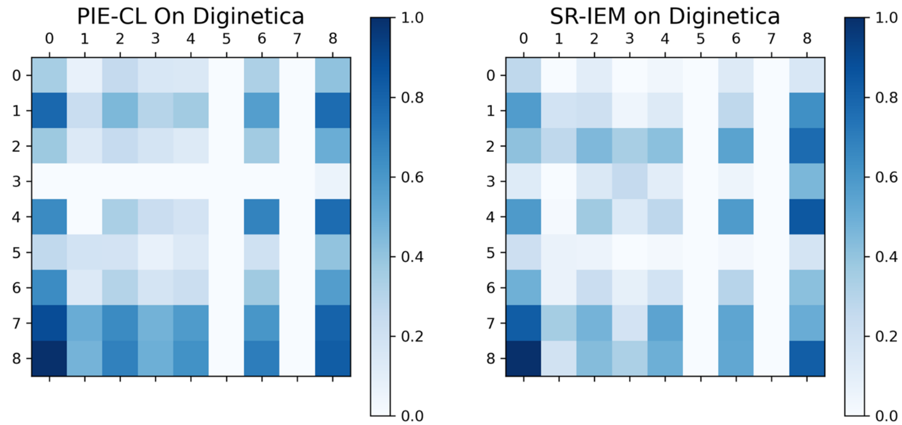
Session length is an important factor influencing the recommendation performance (Wang et al. 2021). Generally, the longer the session, the poorer the recommendation performance. When the session sequence is short, the importance of location information is not obvious. However, as the session length increases, the methods without special design such as SR-IEM cannot capture the sequential information. Here we add supplementary experiments to verify this hypothesis. Figure 11 shows prediction performance varying with different session lengths. This result is consistent with our motivation that the disadvantage of disregarding temporal information is magnified as the length of the sequence increases. From Fig. 12, we can observe that our model using the position-coding mechanism is more robust to session clip length, while the performance of SR-IEM drops as the clipping threshold increases. Both experimental results suggest that the position-coding mechanism can capture the relative position information of items in the sequence, enabling our model to deal with long sessions.

5.10 Parameter sensitivity (Q5)

We now study how two key hyperparameters affect the recommendation performance of our methods. As shown in Fig. 13, we fix other parameters as the default values and vary λ and b in the ranges of {0, 1, 5, 10, 15} and {64, 128, 256, 512, 1024} on



(a) PIE-CL (1/64). **(b)** SR-IEM (1/64).



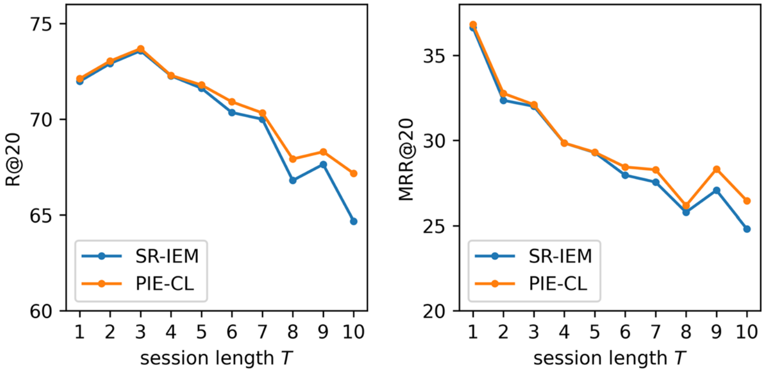
(c) PIE-CL (Diginetica). **(d)** SR-IEM (Diginetica).

Fig. 10 Visualization of the affinity matrix

Yoochoose. Figure 13 shows that our methods achieve best performance when $\lambda = 10$ and $B = 128$.

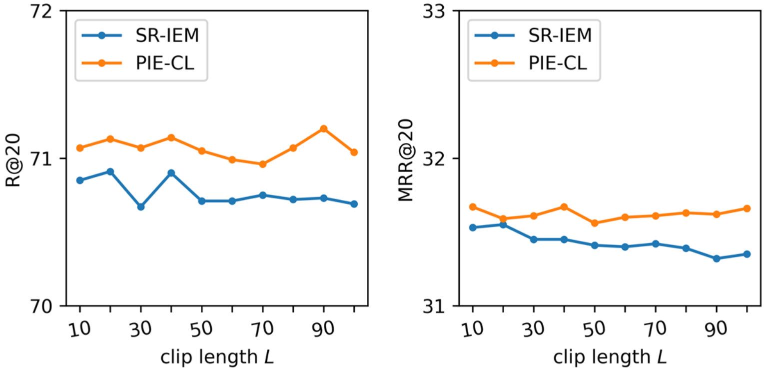
5.11 Model efficiency

Next, we investigate the model efficiency by comparing PIE-CL with several state-of-the-art baselines. Table 9 summarizes the computational complexity as well as the training and testing time of different algorithms. Item-KNN considers only the last interaction in a session and recommends the most similar items according to



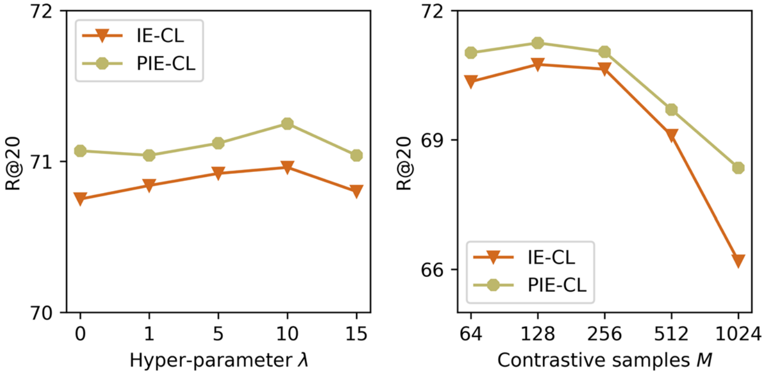
(a) HR@20. **(b)** MRR@20.

Fig. 11 Influence of session length (Yoochoose1/64)



(a) HR@20. **(b)** MRR@20.

Fig. 12 Performance comparison by varying the length of session truncation (Yoochoose1/64)



(a) Effect of λ. **(b)** Effect of M.

Fig. 13 Influence of parameters (Yoochoose1/64)

Table 9 Model complexity and efficiency comparisons

Method	Complexity	Yochoose 1/64		Diginetica	
		Train	Test	Train	Test
Item-KNN	$O(n^2)$	–	1.62	–	1.65
SKNN	$O(n^2S^2)$	–	3.28		–3.40
STAN	$O(tn^2S^2)$	–	3.30	–	–3.44
NARM	$O(td^2+nd^2)$	2.91	2.24	2.41	2.13
STAMP	$O(td^2+nd^2)$	1.19	1.00	1.16	1.00
CSRM	$O(td^2+dN+nd^2)$	4.91	18.62	4.63	19.32
SR-GNN	$O(s(td^2+t^3)+nd^2)$	3.12	2.89	2.56	2.75
RepeatNet	$O(td^2+nd^2)$	2.40	1.76	2.22	1.68
SR-IEM	$O(t^2d+nd^2)$	1.00 (0.5h)	1.00(0.01h)	1.00(1.4h)	1.00(0.01h)
PIE-CL	$O(t^2d+B^2d+nd^2)$	1.25	1.00	1.20	1.00

S: the number of session in the dataset; t: the session length; d: the embedding dimensions; n: the size of vocabulary set; N: the size of memory bank; s: the training steps in GNN, B: the batch size. Note that heuristic methods, i.e., Item-KNN, SKNN, and STAN, do not need to learn models and only create internal memory indexing data structures in the training phase. Compared to neural methods, the time spent in the training phase of heuristic methods is trivial and can be neglected

their co-occurrence in other sessions. Compared to Item-KNN, SKNN requires an extra similarity computation between any two sessions. STAN is a more complex model because it needs to calculate weights for each position in a session. The results generally show that the complexity of deep models is, as expected, much higher than non-neural approaches. However, the prediction times of nearest-neighbors-based methods are often slightly higher than deep-learning-based models.

For NARM, STAMP, and RepeatNet, the main computation overhead is attention calculation and similarity matching. CSRM and SR-GNN require longer computing time and more memory consumption—the former needs to interact with an external memory constantly, while the latter requires the construction of graph and learning interactions with a graph neural network. The main computation of SR-IEM is the importance extraction module (Pan et al. 2020b) and is the most efficient method. Compared to SR-IEM, the extra overhead of PIE-CL comes from the contrastive-based auxiliary task, which requires $O(B^2d)$ time to generate positive and negative representations. As $B \ll n$, the additional consumption from the auxiliary task is trivial.

5.12 Limitations

We have conducted comprehensive experiments to evaluate the proposed PIE-CL model. The experimental results indicate that our model is promising in addressing the SBRS problem. However, there are also several limitations inherent in the proposed model that need to be further examined in the future.

Table 10 Results on five subsets splitting of Diginetica

Method	Year	Diginetica		
		HR@20	MRR@20	Cov@20
SKNN	2017	0.4748	0.1714	0.8701
STAN	2019	0.4803	0.1837	0.9384
GRU4REC	2016	0.4639	0.1644	0.9498
NARM	2017	0.4188	0.1392	0.8696
STAMP	2018	0.3917	0.1314	0.9188
CSRM	2019	0.4258	0.1421	0.7337
SR-GNN	2019	0.3638	0.1564	0.8593
PIE-CL	–	0.4669	0.1663	0.9110

First, the performance gain of PIE-CL over previous methods varied in different datasets. For example, PIE-CL achieves statistically significant improvement on Diginetica, but the discrepancies between PIE-CL and other models are relatively small in Yoochoose. We hypothesize that this phenomenon happens due to dataset biases, which are difficult to avoid but can lead to poor generalization performance. Therefore, how to further improve the proposed method while generalizing its performance in datasets with different distributions is of great interest for future works.

Second, the experimental setting follows the typical studies (Li et al. 2017; Liu et al. 2018; Garg et al. 2019; Ren et al. 2019; Wu et al. 2019) for session-based recommendation. However, the evaluations are conducted on datasets using a single time-ordered training-testing split, which, however, may lead to undesired random effects (Ludewig et al. 2021). We note that there is a different data splitting method (Ludewig et al. 2021) that creates five non-overlapping and contiguous subsets (splits) of the datasets. For each subset, the data of the last seven days is used for testing, and the other data is used for model training. The final results are obtained by averaging the tested results of five subsets. To evaluate our model in such an experimental setting, we conducted an extra experiment on Diginetica strictly following the optimized hyperparameters reported in Ludewig et al. (2021) and the corresponding Github repository[4].

Table 10 summarizes the performance comparisons, which demonstrate that our model yields the best results on HR@20 and MRR@20 compared to deep learning models. However, traditional heuristic methods such as SKNN and STAN show better performance in this experimental setting. This result is consistent with a recent study (Ludewig et al. 2021) and raises an interesting problem, i.e., why the performance of deep recommendation models (including ours) drop more apparently than the simple methods. One possible reason is that the training data in each subset is considerably less (∼1/5) than the data used in Li et al. (2017), Liu et al. (2018), Garg et al. (2019), Ren et al. (2019) and Wu et al. (2019). It is well-known that deep recommendation models require more training data, but their performance is constrained by the smaller data in such a setting. Besides, the potential associations of the events captured by the deep learning models may not reflect users' real intent due to the flexible-ordered user behaviors in a session. Although the position encoding method in our model can

[4] https://github.com/rn5l/session-rec

(to some extent) alleviate this issue, its performance is also limited by the small size of training data. Explaining the model behavior and the recommendation results in different settings is beyond the scope of this study and left as our future work.

Finally, self-supervised pre-training might not be the optimal solution (He et al. 2019) for models relying on latent representation learning such as STAMP and SR-IEM, as pre-training will freeze the majority of the parameters, i.e., it leaves less space for self-supervised pre-training to further improve the model performance. Under such circumstances, joint training might be a good choice (Yang et al. 2019; Wei et al. 2020; Wen et al. 2020; Bingel and Søgaard 2017) that may regularize the model parameters during training. Besides, our method requires more negative samples to improve the performance and robustness of contrastive learning, which is a major drawback of self-supervised learning models, as has been proved in recent theoretical works (Tian et al. 2020; He et al. 2020). Therefore, it is of interest to increase the number of negative samples using methods such as CMC (Tian et al. 2020) and MoCo (He et al. 2020), or even adjust the weights among multiple losses adaptively (Kendall et al. 2018), which, however, are beyond the scope of this paper and left as our future work.

6 Conclusions

In this paper, we proposed PIE-CL, a novel self-supervised session-based recommendation model that explores the additional signals using contrastive learning. We introduced a new auxiliary training objective to distill information for predicting next interaction from the reverse matching paradigm, as well as a bi-directional matching algorithm with contrastive loss, which allow us to discover potential patterns through maximizing the mutual information. Additionally, we designed a simple yet effective learnable position-coding mechanism that can learn user-item interactive patterns adaptively. Extensive experiments conducted on two real-world datasets demonstrated the effectiveness of our model. In the future, we are interested in incorporating more negative samples without the constraints of batch size by introducing external memory bank. Moreover, adaptive weight adjustment is also of interest that may further improve the sequential recommendation performance.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62176043 and No. 62072077).

Appendix

Notations

Table 11 Frequently used notations

Notation	Description
Θ	Parameters in PIE-CL
$\mathcal{F}$	Activation function
B	The batch size
T	The session's length
K	K value in objective evaluations
L	Clip value for session truncation
N	The size of the memory bank
$\mathcal{S}/x_i$	Session set / i-th item
$\mathcal{V}/v_i$	Item embedding set / i-th item
t	The last time step of the session
n	The total number of items in the item set $\mathcal{V}$
$\mathbf{e}_i$	The embedding of i-th item
$\mathbf{p}_i$	The embedding of position for i-th item
$\mathcal{A}$	The affinity matrix for importance extraction
$\mathbf{Q}$	The query matrix for importance extraction
$\mathbf{K}$	The key matrix for importance extraction
d	The dimensionality of hidden state
α_i	The attention weight of i-th item
$\mathbf{z}_l$	The long-term preference
$\mathbf{z}_s$	The short-term preference
$\mathbf{z}$	The individual preference
$\hat{z}_i$	The matching score with a certain item
$\hat{y}_i$	The matching probability with a certain item
$\mathcal{L}$	The cross-entropy loss
$\mathcal{L}^i_{CL}$	The proposed contrastive learning loss for i-th record
$\mathbf{z}_i^+$	The mean vector of positive session representation
$\mathbf{z}_i^-$	The mean vector of negative session representation
$\mathbf{x}_{it}$	The target representation for i-th record
M	The sum of contrastive samples
x	The simplified representation for target item
x^+	The simplified representation for matching score
x^-	The simplified representation for non-matching score
τ	The temperature parameter
λ	The hyper-parameter for losses compromise

Vectors are denoted by boldface lowercase letters, Matrices are denoted by boldface uppercase letters

Table 12 HR@K results on Yoochoose1/64, $K = [1, 3, 5, 10, 15, 20]$

Method	Yoochoose1/64					
	HR@1	HR@3	HR@5	HR@10	HR@15	HR@20
STAMP	0.1722	0.3678	0.4675	0.5951	0.6603	0.6995
STAMP-CL	0.1777	0.3682	0.4689	0.5967	0.6624	0.7011
RepeatNet	0.1831	0.3704	0.4732	0.6001	0.6677	0.7071
RepeatNet-CL	0.1846	0.3710	0.4737	0.6015	0.6689	0.7104
SR-GNN	0.1778	0.3732	0.4756	0.6028	0.6662	0.7061
SR-GNN-CL	0.1777	0.3730	0.4746	0.6013	0.6659	0.7051
SR-IEM	0.1849	0.3762	0.4710	0.6032	0.6671	0.7075
SR-IEM-CL	0.1851	0.3772	0.4796	0.6057	0.6699	0.7101
PIE	0.1848	0.3787	0.4794	0.6066	0.6712	0.7107
PIE-CL	0.1856	0.3792	0.4809	0.6090	0.6727	0.7125

Table 13 MRR@K results on Yoochoose1/64, $K = [1, 3, 5, 10, 15, 20]$

Method	Yoochoose1/64					
	MRR@1	MRR@3	MRR@5	MRR@10	MRR@15	MRR@20
STAMP	0.1722	0.2565	0.2793	0.2965	0.3017	0.3039
STAMP-CL	0.1777	0.2597	0.2827	0.2999	0.3052	0.3074
RepeatNet	0.1831	0.2639	0.2874	0.3047	0.3099	0.3103
RepeatNet-CL	0.1846	0.2648	0.2882	0.3054	0.3107	0.3131
SR-GNN	0.1778	0.2617	0.2851	0.3022	0.3072	0.3095
SR-GNN-CL	0.1777	0.2616	0.2849	0.3020	0.3072	0.3094
SR-IEM	0.1849	0.2672	0.2903	0.3075	0.3125	0.3145
SR-IEM-CL	0.1851	0.2678	0.2912	0.3083	0.3133	0.3151
PIE	0.1848	0.2673	0.2914	0.3084	0.3144	0.3156
PIE-CL	0.1856	0.2689	0.2929	0.3095	0.3145	0.3170

Table 14 nDCG@K results on Yoochoose1/64, $K = [1, 3, 5, 10, 15, 20]$

Method	Yoochoose1/64					
	nDCG@1	nDCG@3	nDCG@5	nDCG@10	nDCG@15	nDCG@20
STAMP	0.1722	0.2850	0.3261	0.3675	0.3848	0.3941
STAMP-CL	0.1777	0.2875	0.3289	0.3705	0.3879	0.3971
RepeatNet	0.1839	0.2912	0.3335	0.3752	0.3927	0.4018
RepeatNet-CL	0.1846	0.2919	0.3342	0.3757	0.3936	0.4034
SR-GNN	0.1778	0.2902	0.3324	0.3737	0.3905	0.4002
SR-GNN-CL	0.1777	0.2901	0.3320	0.3732	0.3906	0.4001
SR-IEM	0.1849	0.2951	0.3368	0.3781	0.3949	0.4043
SR-IEM-CL	0.1851	0.2959	0.3381	0.3790	0.3951	0.4054
PIE	0.1848	0.2956	0.3371	0.3791	0.3962	0.4052
PIE-CL	0.1856	0.2971	0.3391	0.3807	0.3974	0.4069

Table 15 Coverage@K results on Yoochoose 1/64, $K = [1, 3, 5, 10, 15, 20]$

Method	Yoochoose1/64					
	COV@1	COV@3	COV@5	COV@10	COV@15	COV@20
STAMP	0.1134	0.1782	0.2109	0.2529	0.2761	0.2906
STAMP-CL	0.1178	0.1866	0.2219	0.2661	0.2893	0.3038
RepeatNet	0.1127	0.1686	0.1989	0.2324	0.2508	0.2628
RepeatNet-CL	0.1223	0.1825	0.2092	0.2439	0.2640	0.2773
SR-GNN	0.1194	0.1902	0.2251	0.2745	0.3002	0.3165
SR-GNN-CL	0.1188	0.1890	0.2241	0.2714	0.2978	0.3132
SR-IEM	0.1147	0.1883	0.2231	0.2694	0.2944	0.3099
SR-IEM-CL	0.1165	0.1892	0.2243	0.2712	0.2958	0.3109
PIE	0.1175	0.1877	0.2214	0.2684	0.2939	0.3114
PIE-CL	0.1191	0.1894	0.2231	0.2707	0.2990	0.3149

Table 16 HR@K results on Yoochoose1/4, $K = [1, 3, 5, 10, 15, 20]$

Method	Yoochoose1/4					
	HR@1	HR@3	HR@5	HR@10	HR@15	HR@20
STAMP	0.1766	0.3678	0.4700	0.5994	0.6643	0.7046
STAMP-CL	0.1774	0.3722	0.4707	0.6008	0.6652	0.7068
RepeatNet	0.1855	0.3750	0.4780	0.6075	0.6735	0.7150
RepeatNet-CL	0.1880	0.3771	0.4787	0.6087	0.6755	0.7184
SR-GNN	0.1889	0.3807	0.4820	0.6093	0.6737	0.7136
SR-GNN-CL	0.1901	0.3810	0.4817	0.6096	0.6741	0.7147
SR-IEM	0.1857	0.3777	0.4811	0.6091	0.6739	0.7135
SR-IEM-CL	0.1849	0.3783	0.4814	0.6111	0.6751	0.7148
PIE	0.1869	0.3794	0.4842	0.6113	0.6756	0.7162
PIE-CL	0.1876	0.3804	0.4852	0.6152	0.6788	0.7183

Table 17 MRR@K results on Yoochoose1/4, $K = [1, 3, 5, 10, 15, 20]$

Method	Yoochoose1/4					
	MRR@1	MRR@3	MRR@5	MRR@10	MRR@15	MRR@20
STAMP	0.1766	0.2590	0.2823	0.2997	0.3049	0.3071
STAMP-CL	0.1774	0.2612	0.2837	0.3012	0.3063	0.3086
RepeatNet	0.1855	0.2655	0.2916	0.3070	0.3123	0.3147
RepeatNet-CL	0.1880	0.2683	0.2929	0.3097	0.3151	0.3175
SR-GNN	0.1889	0.2713	0.2945	0.3116	0.3167	0.3189
SR-GNN-CL	0.1901	0.2723	0.2953	0.3125	0.3176	0.3199
SR-IEM	0.1857	0.2681	0.2918	0.3090	0.3141	0.3163
SR-IEM-CL	0.1849	0.2681	0.2917	0.3091	0.3142	0.3170
PIE	0.1869	0.2667	0.2916	0.3087	0.3138	0.3171
PIE-CL	0.1876	0.2697	0.2937	0.3110	0.3160	0.3190

Table 18 nDCG@K results on Yoochoose1/4, $K = [1, 3, 5, 10, 15, 20]$

Method	Yoochoose1/4					
	nDCG@1	nDCG@3	nDCG@5	nDCG@10	nDCG@15	nDCG@20
STAMP	0.1766	0.2869	0.3289	0.3709	0.3889	0.3976
STAMP-CL	0.1774	0.2896	0.3302	0.3724	0.3895	0.3993
RepeatNet	0.1855	0.2932	0.3361	0.3792	0.3969	0.4063
RepeatNet-CL	0.1880	0.2959	0.3383	0.3804	0.3984	0.4086
SR-GNN	0.1889	0.2993	0.3410	0.3823	0.3994	0.4067
SR-GNN-CL	0.1901	0.3002	0.3416	0.3831	0.4002	0.4081
SR-IEM	0.1857	0.2962	0.3388	0.3803	0.3974	0.4059
SR-IEM-CL	0.1849	0.2963	0.3388	0.3808	0.3978	0.4071
PIE	0.1869	0.2958	0.3389	0.3782	0.3962	0.4068
PIE-CL	0.1876	0.2978	0.3410	0.3806	0.3994	0.4091

Table 19 Coverage@K results on Yoochoose1/4, $K = [1, 3, 5, 10, 15, 20]$

Method	Yoochoose1/4					
	COV@1	COV@3	COV@5	COV@10	COV@15	COV@20
STAMP	0.1179	0.1891	0.2299	0.2838	0.3134	0.3332
STAMP-CL	0.1184	0.1920	0.2321	0.2847	0.3183	0.3393
RepeatNet	0.1239	0.1887	0.2192	0.2633	0.2885	0.3061
RepeatNet-CL	0.1243	0.1889	0.2202	0.2643	0.2908	0.3075
SR-GNN	0.1230	0.1947	0.2335	0.2870	0.3176	0.3378
SR-GNN-CL	0.1232	0.1967	0.2342	0.2898	0.3204	0.3412
SR-IEM	0.1151	0.1825	0.2195	0.2711	0.3007	0.3196
SR-IEM-CL	0.1139	0.1821	0.2189	0.2704	0.3008	0.3195
PIE	0.1135	0.1845	0.2212	0.2741	0.3039	0.3228
PIE-CL	0.1142	0.1849	0.2227	0.2743	0.3046	0.3236

Table 20 HR@K results on Diginetica, $K = [1, 3, 5, 10, 15, 20]$

Method	Diginetica					
	HR@1	HR@3	HR@5	HR@10	HR@15	HR@20
STAMP	0.0856	0.1962	0.2675	0.3821	0.4579	0.5126
STAMP-CL	0.0875	0.1975	0.2688	0.3857	0.4600	0.5152
RepeatNet	0.0771	0.1820	0.2532	0.3707	0.4461	0.5020
RepeatNet-CL	0.0795	0.1871	0.2854	0.3800	0.4580	0.5165
SR-GNN	0.0878	0.1918	0.2617	0.3757	0.4507	0.5037
SR-GNN-CL	0.0890	0.1970	0.2684	0.3834	0.4567	0.5103
SR-IEM	0.0829	0.1878	0.2596	0.3768	0.4540	0.5100
SR-IEM-CL	0.0844	0.1924	0.2618	0.3808	0.4570	0.5135
PIE	0.0861	0.1928	0.2657	0.3821	0.4589	0.5154
PIE-CL	0.0882	0.1976	0.2707	0.3887	0.4652	0.5220

Table 21 MRR@K results on Diginetica, $K = [1, 3, 5, 10, 15, 20]$

Method	Diginetica					
	MRR@1	MRR@3	MRR@5	MRR@10	MRR@15	MRR@20
STAMP	0.0856	0.1339	0.1501	0.1653	0.1712	0.1743
STAMP-CL	0.0875	0.1356	0.1517	0.1681	0.1731	0.1762
RepeatNet	0.0771	0.1215	0.1377	0.1532	0.1592	0.1623
RepeatNet-CL	0.0795	0.1252	0.1413	0.1574	0.1636	0.1668
SR-GNN	0.0878	0.1321	0.1480	0.1631	0.1690	0.1719
SR-GNN-CL	0.0890	0.1353	0.1514	0.1667	0.1724	0.1755
SR-IEM	0.0829	0.1277	0.1440	0.1595	0.1656	0.1687
SR-IEM-CL	0.0844	0.1305	0.1463	0.1620	0.1680	0.1711
PIE	0.0861	0.1315	0.1481	0.1635	0.1695	0.1727
PIE-CL	0.0882	0.1351	0.1517	0.1674	0.1734	0.1770

Table 22 nDCG@K results on Diginetica, $K = [1, 3, 5, 10, 15, 20]$

Method	Diginetica					
	nDCG@1	nDCG@3	nDCG@5	nDCG@10	nDCG@15	nDCG@20
STAMP	0.0856	0.1499	0.1791	0.2161	0.2361	0.2490
STAMP-CL	0.0875	0.1523	0.1804	0.2189	0.2386	0.2512
RepeatNet	0.0771	0.1370	0.1662	0.2041	0.2240	0.2372
RepeatNet-CL	0.0795	0.1410	0.1702	0.2094	0.2301	0.2439
SR-GNN	0.0878	0.1474	0.1761	0.2128	0.2327	0.2450
SR-GNN-CL	0.0890	0.1511	0.1803	0.2174	0.2368	0.2495
SR-IEM	0.0829	0.1431	0.1725	0.2103	0.2308	0.2437
SR-IEM-CL	0.0844	0.1464	0.1748	0.2132	0.2333	0.2463
PIE	0.0861	0.1472	0.1771	0.2146	0.2350	0.2483
PIE-CL	0.0882	0.1511	0.1811	0.2192	0.2394	0.2528

Table 23 Coverage@K results on Diginetica, $K = [1, 3, 5, 10, 15, 20]$

Method	Diginetica					
	COV@1	COV@3	COV@5	COV@10	COV@15	COV@20
STAMP	0.3407	0.5494	0.6583	0.8026	0.8715	0.9109
STAMP-CL	0.3573	0.5562	0.6677	0.8124	0.8831	0.9216
RepeatNet	0.2866	0.4636	0.5549	0.6856	0.7585	0.8060
RepeatNet-CL	0.3121	0.5170	0.6174	0.7562	0.8304	0.8776
SR-GNN	0.2938	0.4830	0.5757	0.6952	0.7518	0.8142
SR-GNN-CL	0.3228	0.5304	0.6364	0.7813	0.8514	0.8858
SR-IEM	0.2973	0.5051	0.6107	0.7507	0.8256	0.8708
SR-IEM-CL	0.3202	0.5361	0.6482	0.7974	0.8706	0.9130
PIE	0.3034	0.5079	0.6151	0.7586	0.8334	0.8792
PIE-CL	0.3294	0.5438	0.6565	0.8060	0.8794	0.9194

References

Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., Saunshi, N.: A theoretical analysis of contrastive unsupervised representation learning. In: 36th International Conference on Machine Learning, ICML 2019, pp. 9904–9923. International Machine Learning Society (IMLS) (2019)

Bai, T., Nie, J.Y., Zhao, W.X., Zhu, Y., Du, P., Wen, J.R.: An attribute-aware neural attentive model for next basket recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 1201–1204 (2018)

Baxter, J.: A model of inductive bias learning. J. Artif. Intell. Res. **12**, 149–198 (2000)

Belghazi, M.I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., Hjelm, R.D.: Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062 (2018)

Benson, A.R., Kumar, R., Tomkins, A.: Modeling user consumption sequences. In: Proceedings of the 25th International Conference on World Wide Web, pp. 519–529 (2016)

Bingel, J., Søgaard, A.: Identifying beneficial task relations for multi-task learning in deep neural networks. In: EACL (2), pp. 164–169. Association for Computational Linguistics (2017)

Bollmann, M., Søgaard, A.: Improving historical spelling normalization with bi-directional lstms and multi-task learning. In: COLING, pp. 131–139. ACL (2016)

Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model. User-Adap. Int. **24**(1), 67–119 (2014)

Caruana, R.: Multitask learning: a knowledge-based source of inductive bias icml. Google Scholar Google Scholar Digital Library Digital Library (1993)

Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)

Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 191–198 (2016)

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., et al.: The youtube video recommendation system. In: Proceedings of the fourth ACM Conference on Recommender Systems, pp. 293–296 (2010)

Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1), pp. 4171–4186. Association for Computational Linguistics (2019)

Garg, D., Gupta, P., Malhotra, P., Vig, L., Shroff, G.: Sequence and time aware neighborhood for session-based recommendations: Stan. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1069–1072 (2019)

Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: AISTATS, pp. 297–304 (2010)

Hariri, N., Mobasher, B., Burke, R.: Adapting to user preference changes in interactive recommendation. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)

He, K., Girshick, R., Dollár, P.: Rethinking imagenet pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4918–4927 (2019)

He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 549–558 (2016)

Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: ICLR (Poster) (2016)

Hidasi, B., Quadrana, M., Karatzoglou, A., Tikk, D.: Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 241–248 (2016)

Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. In: ICLR. OpenReview.net (2019)

Hu, L., Cao, L., Wang, S., Xu, G., Cao, J., Gu, Z.: Diversifying personalized recommendation with user-session context. In: IJCAI, pp. 1858–1864 (2017)

Jannach, D., Ludewig, M.: When recurrent neural networks meet the neighborhood for session-based recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 306–310 (2017)

Ke, G., He, D., Liu, T.Y.: Rethinking positional encoding in language pre-training. In: International Conference on Learning Representations (2020)

Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7482–7491 (2018)

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Adv. Neural Inf. Process. Syst. **33** (2020)

Kong, L., d'Autume, C.D.M., Ling, W., Yu, L., Dai, Z., Yogatama, D.: A mutual information maximization perspective of language representation learning. In: ICLR (2020)

Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)

Kumar, B.G.V., Carneiro, G., Reid, I.: Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5385–5394 (2016)

LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)

Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1419–1428 (2017)

Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.S.: Gated graph sequence neural networks. In: ICLR (Poster) (2016)

Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: Stamp: short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1831–1839 (2018)

Liu, X., Zhang, F., Hou, Z., Wang, Z., Mian, L., Zhang, J., Tang, J.: Self-supervised learning: generative or contrastive. arXiv:2006.08218 (2020)

Ludewig, M., Mauro, N., Latifi, S., Jannach, D.: Empirical analysis of session-based recommendation algorithms. User Model. User-Adap. Int. **31**(1), 149–181 (2021)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR (Workshop) (2013)

Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Using sequential and non-sequential patterns in predictive web usage mining tasks. In: 2002 IEEE International Conference on Data Mining, 2002. Proceedings, pp. 669–672. IEEE (2002)

Niranjan, U., Subramanyam, R., Khanaa, V.: Developing a web recommendation system based on closed sequential patterns. In: International Conference on Advances in Information and Communication Technologies, pp. 171–179. Springer (2010)

Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)

Pan, Z., Cai, F., Chen, W., Chen, H., de Rijke, M.: Star graph neural networks for session-based recommendation. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 1195–1204 (2020)

Pan, Z., Cai, F., Ling, Y., de Rijke, M.: Rethinking item importance in session-based recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1837–1840 (2020)

Qiu, R., Huang, Z., Li, J., Yin, H.: Exploiting cross-session information for session-based recommendation with graph neural networks. ACM Trans. Inf. Syst. (TOIS) **38**(3), 1–23 (2020)

Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P.: Personalizing session-based recommendations with hierarchical recurrent neural networks. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 130–137 (2017)

Ren, P., Chen, Z., Li, J., Ren, Z., Ma, J., de Rijke, M.: Repeatnet: a repeat aware neural recommendation machine for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4806–4813 (2019)

Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461. AUAI Press (2012)
Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820 (2010)
Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295 (2001)
Schmarje, L., Santarossa, M., Schröder, S.M., Koch, R.: A survey on semi, self-and unsupervised techniques in image classification. arXiv:2002.08721 (2020)
Shani, G., Heckerman, D., Brafman, R.I., Boutilier, C.: An mdp-based recommender system. J. Mach. Learn. Res. **6**(9) (2005)
Socher, R., Lin, C.C., Manning, C., Ng, A.Y.: Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 129–136 (2011)
Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. Adv. Neural. Inf. Process. Syst. **27**, 3104–3112 (2014)
Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 565–573 (2018)
Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: ECCV (11). Lecture Notes in Computer Science, vol. 12356, pp. 776–794. Springer (2020)
Twardowski, B.: Modelling contextual information in session-aware recommender systems with neural networks. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 273–276 (2016)
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
Velickovic, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. In: ICLR (Poster) (2019)
Wang, M., Ren, P., Mei, L., Chen, Z., Ma, J., de Rijke, M.: A collaborative session-based recommendation approach with parallel memory modules. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 345–354 (2019)
Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., Cheng, X.: Learning hierarchical representation model for nextbasket recommendation. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 403–412 (2015)
Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M., Lian, D.: A survey on session-based recommender systems. ACM Comput. Surv. (2021)
Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. In: IJCAI, pp. 4144–4150. ijcai.org (2017)
Wei, H., Feng, L., Chen, X., An, B.: Combating noisy labels by agreement: a joint training method with co-regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13726–13735 (2020)
Wen, H., Zhang, J., Wang, Y., Lv, F., Bao, W., Lin, Q., Yang, K.: Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2377–2386 (2020)
Wu, C., Yan, M.: Session-aware information embedding for e-commerce product recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge management, pp. 2379–2382 (2017)
Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 346–353 (2019)
Xu, C., Zhao, P., Liu, Y., Sheng, V.S., Xu, J., Zhuang, F., Fang, J., Zhou, X.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI, vol. 19, pp. 3940–3946 (2019)
Xu, Y., Chen, J., Huang, C., Zhang, B., Xing, H., Dai, P., Bo, L.: Joint modeling of local and global behavior dynamics for session-based recommendation. In: ECAI 2020, pp. 545–552. IOS Press (2020)

Yang, Z., Cheng, Y., Liu, Y., Sun, M.: Reducing word omission errors in neural machine translation: A contrastive learning approach. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6191–6196 (2019)

Yu, F., Zhu, Y., Liu, Q., Wu, S., Wang, L., Tan, T.: Tagnn: Target attentive graph neural networks for session-based recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1921–1924 (2020)

Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X.: A simple convolutional generative network for next item recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 582–590 (2019)

Zhang, C., Li, Y., Du, N., Fan, W., Yu, P.S.: Entity synonym discovery via multipiece bilateral context matching. In: IJCAI (2020)

Zhang, S., Tay, Y., Yao, L., Sun, A., An, J.: Next item recommendation with self-attentive metric learning. In: Thirty-Third AAAI Conference on Artificial Intelligence, vol. 9 (2019)

Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: a survey and new perspectives. ACM Comput. Surv. (CSUR) **52**(1), 1–38 (2019)

Zhang, Y., Yang, Q.: A survey on multi-task learning. IEEE Trans. Knowl. Data Eng. (2021)

Zhou, C., Ma, J., Zhang, J., Zhou, J., Yang, H.: Contrastive learning for debiased candidate generation in large-scale recommender systems. arXiv:cs.IR/2005.12964 (2020)

Zhou, F., Cao, C., Zhong, T., Geng, J.: Learning meta-knowledge for few-shot image emotion recognition. Expert Syst. Appl. 114274 (2021)

Zhou, F., Wang, P., Xu, X., Tai, W., Trajcevski, G.: Contrastive trajectory learning for tour recommendation. ACM Trans. Intell. Syst. Technol. (2021)

Zhou, F., Xu, X., Trajcevski, G., Zhang, K.: A survey of information cascade analysis: models, predictions, and recent advances. ACM Comput. Surv. **54**(2), 27:1-27:36 (2021)

Zhou, F., Yang, Q., Zhong, T., Chen, D., Zhang, N.: Variational graph neural networks for road traffic prediction in intelligent transportation systems. IEEE Trans. Ind. Inf. **17**(4), 2802–2812 (2021). https://doi.org/10.1109/TII.2020.3009280

Wenxin Tai received B.S. degree in the school of information and communication engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2019, where he is currently pursuing a M.S. degree in software engineering at UESTC. His research interests include recommender systems, speech enhancement and financial technology.

Tian Lan is currently an Associate Professor with the School of Information and Software Engineering, UESTC. His current research interests include medical image processing, speech enhancement, and natural language processing.

Zufeng Wu received the B.S., M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, in 2002, 2006 and 2012, respectively. His research interests include social network data mining and knowledge discovery.

Pengyu Wang is currently working toward a master's degree in the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interests include deep learning and data mining with a special focus on urban computing and spatialtemporal data.

Yixiang Wang is now pursuing M.S. degree in the School of Information and Software Engineering at the University of Electronic Science and Technology of China (UESTC). His research interests include recommender systems and speech enhancement.

Fan Zhou received the B.S. degree in computer science from Sichuan University, China, in 2003, and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, in 2006 and 2012, respectively, where he is currently an Associate Professor with School of Information and Software Engineering. His research interests include machine learning, neural networks, spatio-temporal data management, graph learning, recommender systems, and social network data mining and knowledge discovery.